NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

LINEAR MODELING OF TILTROTOR AIRCRAFT
(IN HELICOPTER AND AIRPLANE MODES)
FOR
STABILITY ANALYSIS AND PRELIMINARY
DESIGN

by

Gary D. Klein

June, 1996

Thesis Co-Advisors: Robert G. Hutchins
                                       E. Roberts Wood

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 3

19961024 016

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE LINEAR MODELING OF TILTROTOR AIRCRAFT (IN HELICOPTER AND AIRPLANE MODES) FOR STABILITY ANALYSIS AND PRELIMINARY DESIGN	5. FUNDING NUMBERS
6. AUTHOR(S) Klein, Gary D.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis investigates the linear state space modeling of a tiltrotor aircraft by modifying an existing MATLAB routine which is used for preliminary (helicopter) stability and control analysis. The modifications consist of changing existing script files along with adding new ones. The modifications result in having a routine that allows the input of tiltrotor characteristics and subsequently generates a state space model along with other stability and control characteristics. The tiltrotor modeling is validated by the input of XV-15 characteristic data into the program and performing a eigenvalue comparison with a model of a similar tiltrotor, the V-22. A more extensive comparison is performed with another XV-15 model which has been extensively used and validated with wind tunnel and flight tests.

14. SUBJECT TERMS XV-15, V-22, Tiltrotor, Linear, State Space, Stability, Control, Rotorcraft, Modeling	15. NUMBER OF PAGES 223
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

i

LINEAR MODELING OF TILTROTOR AIRCRAFT
(IN HELICOPTER AND AIRPLANE MODES)
FOR
STABILITY ANALYSIS AND PRELIMINARY DESIGN

Gary D. Klein
Captain, United States Marine Corps
B.S., United States Naval Academy, 1985
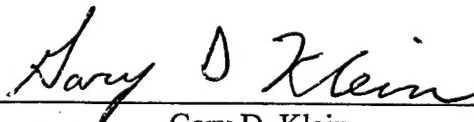
Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
and
MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1996

Author: _____

Gary D. Klein

Approved by: _____

Robert G. Hutchins, Thesis Co-Advisor

_____
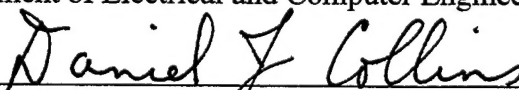
E. Roberts Wood, Thesis Co-Advisor

_____

Herschel H. Loomis, Jr., Chairman
Department of Electrical and Computer Engineering

_____

Daniel J. Collins, Chairman
Department of Aeronautics and Astronautics

iii

ABSTRACT

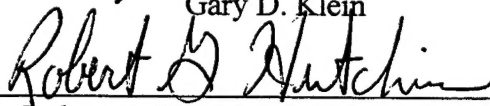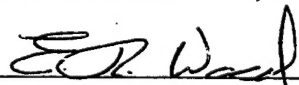This thesis investigates the linear state space modeling of a tiltrotor aircraft by modifying an existing MATLAB routine which is used for preliminary (helicopter) stability and control analysis. The modifications consist of changing existing script files along with adding new ones. The modifications result in having a routine that allows the input of tiltrotor characteristics and subsequently generates a state space model along with other stability and control characteristics. The tiltrotor modeling is validated by the input of XV-15 characteristic data into the program and performing a eigenvalue comparison with a model of a similar tiltrotor, the V-22. A more extensive comparison is performed with another XV-15 model which has been extensively used and validated with wind tunnel and flight tests.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/NOTATION

Text	Code variable name	Definition	Units
A_1	n/a	Lateral blade feathering (tiltrotor equivalent to)	radians
A_w	Aw	Wing reference area	ft^2
A_H	Ah	H-stab reference area	ft^2
A_b	Ab	Blade reference area	ft^2
$a_{1s\,(R\,or\,L)}$	a1s	First harmonic of longitudinal flapping wrt shaft axis	radians
a_o or a	a	Rotorblade lift curve slope, $C_{l_{\alpha(w)}} = \left(\dfrac{\partial C_l}{\partial \alpha}\right)_{blade\ section}$	1/rad
a_w	aw	Wing lift curve slope, $C_{L_{\alpha(w)}} = \left(\dfrac{\partial C_L}{\partial \alpha}\right)_{wing}$	1/rad
a_H	ah	H-stab lift curve slope	1/rad
a_f	af	Fuselage lift curve slope, $C_{l_{\alpha(f)}}$ (referenced to A_w)	1/rad
B_1	n/a	First longitudinal harmonic of blade feathering	radians
$b_{1s\,(R\,or\,L)}$	b1s	First harmonic of lateral flapping wrt shaft axis	radians
c_b	rchord	Blade root aerodynamic chord	ft
c_w	cw	Wing mean aerodynamic chord	ft
c_H	ch	H-stab mean aerodynamic chord	ft
c_V	cv	Vertical Stab mean aerodynamic chord	ft
C_H/σ	chsig	H-force coefficient to solidity ratio	none
C_T/σ	ctsig	Thrust coefficient to solidity ratio	none
C_Q/σ	ctsig	Power coefficient to solidity ratio	none
GW	GW	Aircraft gross weight	lbs
h_m	hm	Rotor hub vertical distance above cg	ft
i_w	iw	Wing incidence angle	radians
i_H	ih	H-stab incidence angle	radians
l_m	lm	Rotor hub longitudinal distance aft of cg	ft
$\mathbf{M}$	n/a	Pitch moment (forcing nose to pitch up)	ft-lbs
$\mathbf{N}$	n/a	Yawing moment (forcing nose yaw to the right)	ft-lbs
p	n/a	Perturbation angular velocity about the X-body axis	rad/sec
q	n/a	Perturbation angular velocity about the Y-body axis	rad/sec
q	q	Airstream dynamic pressure	lbs/ft^2
q_H	N/A	Dynamic pressure @ H-stab	lbs/ft^2

Text	Code variable name	Definition	Units
R	R	Rotor radius	ft
$\mathbf{R}$	n/a	Roll moment (forcing right wing down)	ft-lbs
r	n/a	Perturbation angular vel. about the Z-body axis	rad/sec
$\mathbf{X}$	n/a	Longitudinal body force (positive toward the nose)	lbs
$\mathbf{Y}$	n/a	Sideward body force (positive toward right wing)	lbs
$\mathbf{Z}$	n/a	Vertical body force (positive down)	lbs
$\dot{x}$	n/a	Perturbation velocity in X-body direction	ft/sec
$\dot{y}$	n/a	" " " Y-body "	ft/sec
$\dot{z}$	n/a	" " " Z-body "	ft/sec
y_m	ym	Rotor hub lateral distance from cg	ft
α_w	alpw	Wing angle of attack	radians
α_{olw}	alplow	Wing angle of attack @ zero lift	radians
α_H	N/A	H-stab angle of attack	radians
α_{olf}	alplof	Fuselage angle of attack @ zero lift	radians
δ_e	dele	Elevator angle	radians
ε_o	epso	Downwash angle @ zero (wing) lift	radians
μ	mu	Tip speed ratio, V/V_{tip}	nondim
γ	lockno	Lock number $= \rho a c_b R^4 / I_b$	
ρ	rho	Ambient air density	slugs/ft^3
σ	solidity		
λ'	lamda	Inflow ratio wrt TPP $\lambda' = \mu\alpha_{TPP} - v_1/\Omega R$	nondim
Ω	omega	Rotational velocity of the proprotor	rad/sec
Ψ_B	psi	Aircraft (body axis) yaw angle	radians
Φ_B	phi	Aircraft (body axis) roll angle	radians
θ_B	tho	Aircraft (body axis) pitch angle	radians
θ_o	thetao	Blade (collective) pitch	radians
θ_1	theta1	Blade twist	radians
θ_{oT}	thetaot	Tail rotor pitch or Yaw input (tiltrotor equivalent to) (actually ΔB_1 or $B_{1(right)} = -B_{1(left)}$)	radians

ACKNOWLEDGMENTS

I wish to give thanks to a number of individuals who have unselfishly supported my research on this thesis. To Dr. Bob Wood, a superb advisor to me not only for my thesis, but also toward my future endeavors in the rotorcraft industry. To Sam Ferguson, for his seemingly limitless knowledge of tiltrotor modeling and simulation (and willingness to share some of it to me). I would especially like to thank him for all the uncompensated time he devoted to getting the data I requested for which I could not have done it without his help.

I would like to give my deepest thanks to my loving wife, Traci, and her family for their continued encouragement and undying support throughout my _extended_ time here. Finally, to my daughters, Jessie and Morgan, who without them and their mother to come home to every day, all this would not have been so important. We're _finally_ done!

I. INTRODUCTION

A. TILTROTOR --- HELICOPTER OR AIRPLANE?

The first truly successful tiltrotor aircraft to be flown and fully tested was the Bell/ NASA XV-15. It demonstrated the hover advantages of a helicopter and the speed advantages of a fixed wing aircraft. Since then, there have been many aircraft requirements that could use these advantages. In designing such an aircraft, the problem of predicting the flying qualities is an important part of the process. Helicopter aerodynamics are in themselves very complicated and are often shunned by the aerodynamicist, but a tiltrotor is not only a helicopter, it is an airplane too.

There are two primary modes of flight that are most important in the dynamic analysis, namely helicopter and airplane modes. These are the most important flight modes because they are the ones in which the aircraft spends the majority of its time. In the helicopter mode, the rotors (shafts) are pointed vertically up. In this mode, the aircraft flies like a helicopter, meaning normal "helicopter" flight control movements result in normal "helicopter" type responses. The airplane mode is where the rotors are pointed forward, and the aircraft flies like a normal airplane.

B. BACKGROUND

The Helicopter Design course at the Naval Postgraduate School has been using software called JANRAD developed by students to assist in the preliminary design of rotorcraft. This software (short for Joint Army-Navy Rotorcraft Analysis and Design) is a series of Matlab script files that takes in physical and aerodynamic characteristics of a helicopter and then generates a performance, stability and control analysis on the input aircraft. This past year's design course was given design requirements that were better served by a tiltrotor. The JANRAD software, however, is geared to analyze a helicopter (or compound helicopter) and can only be used to analyze a tiltrotor with modifications. This project covers the modifications that are necessary to accurately model the dynamics of a tiltrotor aircraft.

C. WHAT IS A TILTROTOR ?

Before the dynamics of an aircraft can be discussed, a physical description of the aircraft, including its flying characteristics, is needed. In this description, the reader is assumed

1

to have working knowledge of basic helicopter and airplane dynamics and the associated engineering terminology of both types of aircraft.

In the helicopter mode, the aircraft's rotor dynamics produce the primary forces. Therefore, the design of the rotor head must be examined. The XV-15 and V-22 have very similar rotor head designs. Compared to a helicopter design, they have three blades on a gimbaled head, which is essentially a teetering three bladed rotor (or proprotor as the tiltrotor community calls them). With this design, there is no flapping hinge offset, and each blade's flapping is affected by the gimbal hinge (like a ball and socket joint). The right rotor rotates in the same direction as a conventional (U.S.) helicopter, which is counter clockwise (CCW) as viewed from above the aircraft. The left rotor rotates in the opposite direction to counter the torque of the right rotor.

The proprotor blades differ from a conventional helicopter blade in that they are highly twisted (~ 40 degrees compared to ~10 degrees for a helicopter). This design is a compromise between hover performance which favors slightly twisted blades and forward flight performance (in the airplane mode) which desires the twist to be more like an airplane propeller (~ 60 degrees). Also unlike a helicopter blade, the proprotor blades have variable thickness and taper.

The control of a tiltrotor is the most interesting aspect of the design. Since the analysis will be broken down into two distinct modes -- Helicopter and Airplane, the control dynamics should be too. They are depicted in Figure 1.1. In helicopter mode, the rotor shafts are pointed upward as a conventional helicopter with zero incidence angle. Longitudinal control (cyclic) is essentially made the same way as a helicopter by tilting each swashplate longitudinally. Like a helicopter, this swashplate movement induces cyclic feathering which in turn produces longitudinal flapping. Also the same as a conventional helicopter is collective control. The collective simply moves the swashplates (each in equal amounts) up and down the rotor shaft, which in turn produce more or less pitch on the blades.

Lateral/directional control is where the real difference comes in. First, lateral cyclic, or rolling control, in a tiltrotor is made through differential collective inputs on each rotor. Lateral cyclic produce equal and opposite swashplate movement along the rotor shafts, producing equal an opposite forces on either sides of the aircraft. Although there is some lateral swashplate movement in the V-22, it is not produced by lateral cyclic. Lateral swashplate movement is for lateral cyclic trim for trimming a roll angle for (sideward) sloped landings or for allowing a level fuselage angle for high crosswind approach/landing conditions. This trim is not directly affected

2
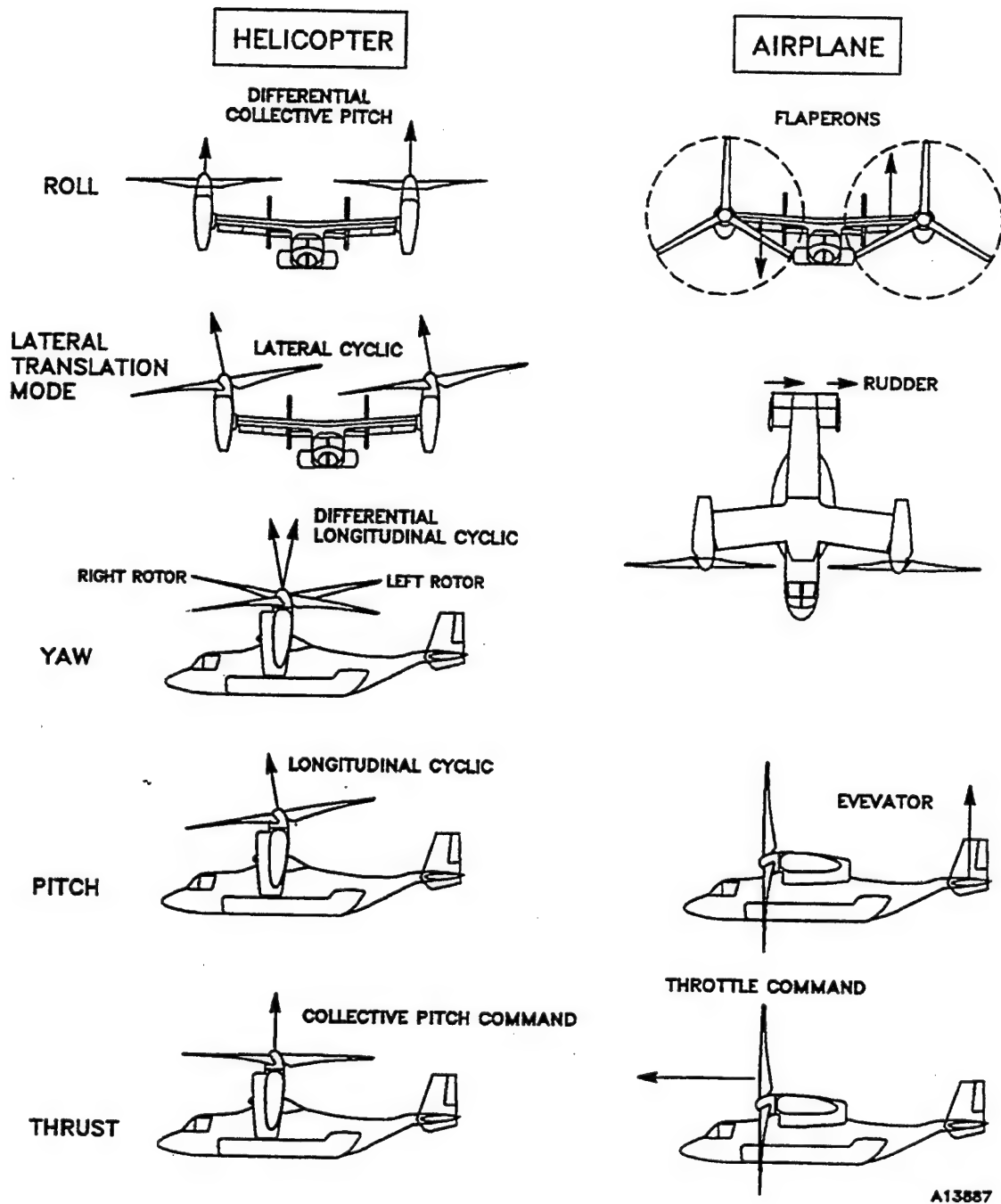
Figure 1.1 Helicopter and Airplane Controls

by lateral control movement, but instead by a trimming thumbwheel on the control stick, which results in the movement of the rotors depicted in Figure 1.1 as the Lateral Translational Mode. Next, directional control is made through differential longitudinal swashplate tilting, which

3

produces a tilting of the thrust vector equal and opposite to the other proprotor head. This produces a yawing moment to facilitate heading changes with the application of the directional pedals.

The airplane mode flight controls are not as interesting or unique. As with most conventional airplanes, longitudinal control is produced through movement of the elevator on the horizontal stabilizer (H-stab). This control input directly produces a pitching moment, M, and indirectly changes the speed of aircraft in trim. The power input in a tiltrotor changes the proprotor thrust, directly changing the speed of the aircraft and indirectly changing the altitude of the aircraft in trim. Lateral control is made through the differential movement of ailerons on the ends of the wing, causing a rolling moment, R. Finally, directional control in the tiltrotor is made through the movement of rudders, as in any twin rudder airplane.

II. SOFTWARE OVERVIEW

A.   ORIGINAL SOFTWARE ARCHITECTURE

JANRAD consists of two major subroutines which are applicable to this project. The first routine, JANRAD Performance, calculates the trim solution and various performance parameters of a helicopter. It is described in detail in Ref. 9. The second one is the JANRAD Stability and Control routine shown in Figure 2.1, and the details of this routine are described in
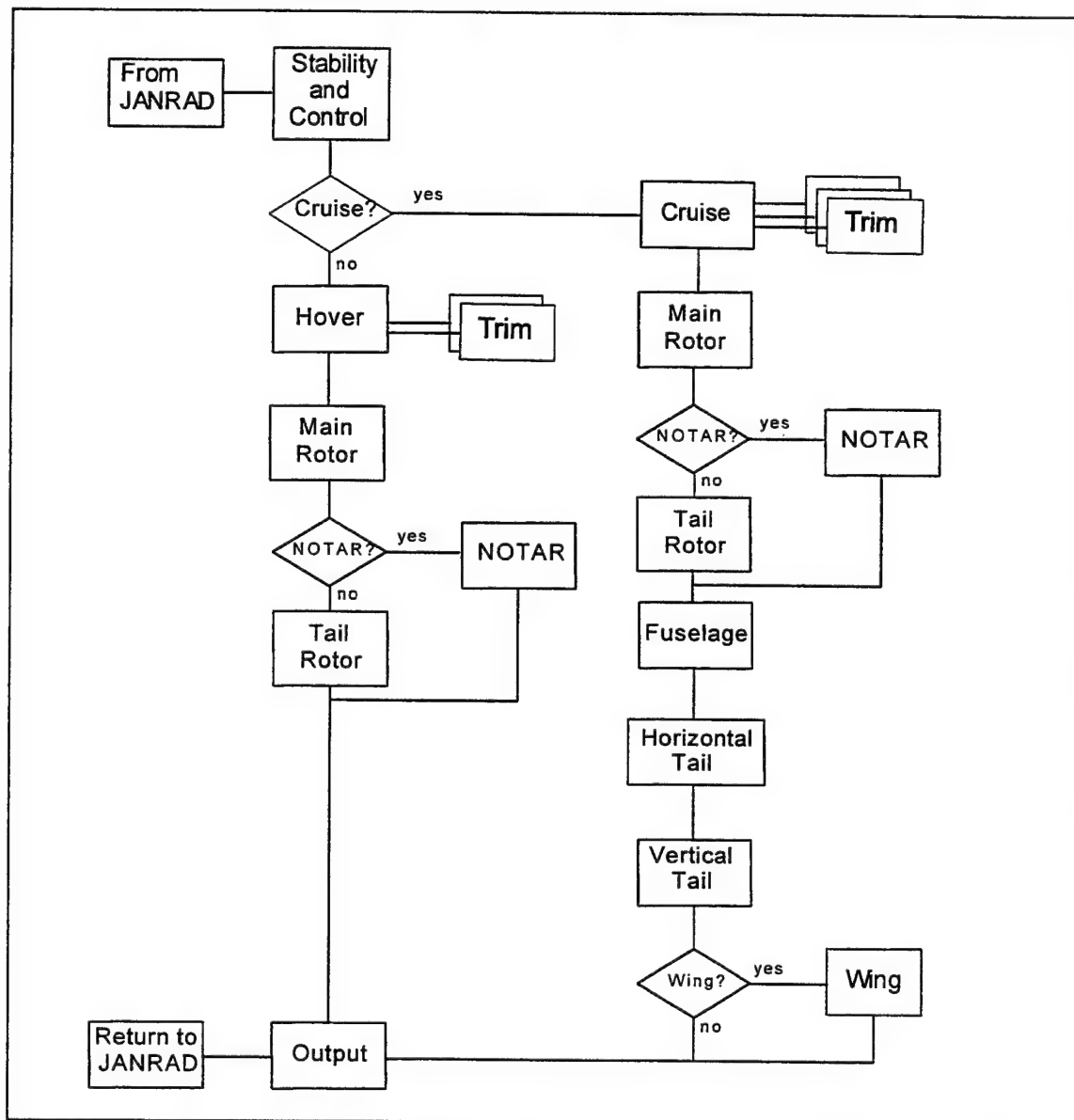
Figure 2.1 JANRAD Architecture

Reference 7. The second will be the main focus of the modifications presented in this project. In this chapter, "JANRAD" will refer to the older version that is to be modified.

The opening menu for JANRAD has the user enter data for performance calculations. If data has already been entered but some need to be modified, it has the following modification menu shown in Figure 2.2. This menu allows for changing all the input parameters needed for the helicopter performance calculations. This set of data is incomplete for stability and control analysis of a tiltrotor. Therefore, an additional set of input menus was developed in order to permit the more detailed data to be entered or changed. A description of these new menus appears in Section C following.

```
                          *** EDIT MENU ***

             1. pressure altitude      2. temperature
             3. airspeed               4. gross weight
           ' 5. number of blades       6. blade radius
             7. blade root chord       8. hinge offset
             9. blade grip length ·   10. blade twist
            11. blade weight          12. # blade elements
            13. rotational velocity   14. # azimuth sectors
            15. lift curve slope      16. airfoil
            17. collective pitch      18. flatplate area
            19. vert projected area   20. wing area
            21. wing span             22. wing CL
            23. wing CDo              24. wing efficiency factor
            25. horizontal tail area  26. horizontal tail span
            27. horizontal tail CL    28. horizontal tail CDo
            29. vertical tail area    30. vertical tail span
            31. vertical tail CL      32. vertical tail CDo
            33. auxiliary thrust      34. rotor blade taper ratio
            35. start of taper

             0. NO CHANGES


         Input the parameter to change:
```

Figure 2.2 JANRAD Main Edit Menu Screen

Once the proper data has been entered, a set of performance calculations must precede the stability calculations in the model. This is because they are based on trimmed flight control positions, and these positions can only be determined through an extensive performance/trim routine. For the helicopter in either hover or forward flight, the trimming involves determining the swashplate position equating to collective pitch (θ_{0M}), longitudinal and lateral flapping angles (a_{1s} and b_{1s} respectively) and (for a conventional helicopter) the collective pitch (θ_{0T}) of the tail rotor. When these are determined, the stability analysis calculations can proceed.

6

B. MODELING CONVENTIONS

The basic aircraft model and modeling conventions of JANRAD are presented here. The coordinate system used in modeling the aircraft is the conventional NACA orthogonal aircraft axis system, where the x-axis runs along the longitudinal axis (directed toward the front) of the aircraft. The y-axis is directed toward the right wing, and the z-axis is perpendicular to x and y, directed downward to the ground. The locations of aircraft components are referenced to a datum position where the fuselage station is the (x) distance aft of the longitudinal datum, buttline is the (y) position right of fuselage centerline and waterline is the vertical location above the (z) datum point.

The simplified dynamic model is nonlinear, meaning the equations of motion are written as a set of nonlinear differential equations. They are a series of three force (one for each axis) and three moment equations. The moment equations were integrated to produce a total of nine equations for this model. These equations were then linearized by using small perturbations about a trimmed (steady-state) condition, retaining the first term of the Taylor Series expansion for each relation. This linearization produces a set of nine constant coefficient, linear differential equations which are expressed in the standard state space format of a matrix differential equation, $d\mathbf{x}/dt = [A]\mathbf{x} + [B]\mathbf{u}$, where the elements in the state vector $\mathbf{x}$ represent the perturbations from the trimmed value in the following order:

$$
\mathbf{x} = \begin{bmatrix} u_B \\ w_B \\ q_B \\ \theta_B \\ v_B \\ p_B \\ \Phi_B \\ r_B \\ \Psi_B \end{bmatrix}
\begin{array}{ll}
\text{velocity of the aircraft in the positive x direction} & \text{(ft/sec)} \\
\text{velocity of the aircraft in the positive z direction} & \text{(ft/sec)} \\
\text{pitch rate of the aircraft (about the positive y - axis)} & \text{(rad/sec)} \\
\text{pitch angle of the aircraft \quad " \quad " \quad " \quad "} & \text{(radians)} \\
\text{velocity of the aircraft in the positive y direction} & \text{(ft/sec)} \\
\text{roll rate of the aircraft (about the positive x - axis)} & \text{(rad/sec)} \\
\text{roll angle of the aircraft \quad " \quad " \quad " \quad "} & \text{(radians)} \\
\text{yaw rate of the aircraft (about the positive z - axis)} & \text{(rad/sec)} \\
\text{yaw angle of the aircraft \quad " \quad " \quad " \quad "} & \text{(radians)}
\end{array}
\tag{2.1}
$$

The first four states are classified as the longitudinal states and the final five are the lateral-directional states. These classifications are used when it is desired to reduce the model to two "uncoupled" state models.

The elements in the control vector **u,** shown below, represent the perturbations in the corresponding feathering of the rotor blades from each of the trimmed control positions.

$$\mathbf{u} \quad = \quad \begin{bmatrix} \delta_{B_1} \\ \delta_{\theta_0} \\ \delta_{A_1} \\ \delta_p \end{bmatrix} \begin{array}{l} \text{longitudinal cyclic (positive aft)} \\ \text{collective pitch \ldots (positive up)} \\ \text{lateral cyclic \ldots (positive right)} \\ \text{directional pedal (positive right)} \end{array} \qquad (2.2)$$

The final control vector used in the model is converted to inches of stick/pedal movement by applying the flight control rigging relationships. The first two inputs produce predominately longitudinal responses, and the second two inputs produce lateral-directional ones.

The [A] matrix contains the resulting coefficients from the Taylor Series first order approximation. These coefficients are simply partial derivatives commonly referred to as the stability derivatives of the aircraft. The stability derivatives make up the [A] matrix as shown in Appendix B. This format is taken from NASA Technical Memorandum 84281 [Ref. 8], which also includes some derivations of the relations used.

C.   SOFTWARE MODIFICATIONS AND ADDITIONS

Because a tiltrotor's flying qualities differ significantly from those of a helicopter, there are different aircraft parameters that affect how the aircraft behaves when perturbed from a trimmed state. Most parameters recorded in JANRAD are the same as those affecting the behavior of a tiltrotor in the hover mode. However, the tiltrotor's dynamics in the airplane mode are affected by parameters not recorded by JANRAD, and it is for this reason that JANRAD's stability input routine, STAB.M, had to be modified. These modifications were made with the fewest number of changes to the routine as possible resulting in JANRAD routines being both modified and augmented by new routines to facilitate the necessary functions. The significantly modified routines are listed in Appendix L and the routines that were written and added to JANRAD are in Appendix M.

As discussed previously, a tiltrotor is similar to a helicopter in the hover mode. The major differences occur in its airplane-like characteristics in the cruise mode. Therefore, in the hover mode, the parameters that JANRAD's stability section uses in its analysis almost suffice for a tiltrotor. The only additional parameter that affects the hover mode dynamics is the addition of a flapping spring constant. Because the tiltrotors have gimbaled rotor heads, their

8

flapping is constrained by centrifugal force and (in some designs) also by a flapping spring which produces significant moments when the blades flap. The first modification is the addition of a flapping spring constant (#6) in the additional parameters change menu screen 1, shown in Figure 2.3.

```
                ***    STABILITY AND CONTROL MENU    ***
                  *** ADDITIONAL PARAMETERS (1 of 3) ***

        Main Rotor
        1. flapping mom of inertia  2. hub height above waterline
        3. hub fuselage station     4. hub posn right of buttline
        5. mast incidence           6. Flapping spring constant

        Tail Rotor (enter zeros (0) if using NOTAR or Tilt Rotor)
        7. height above waterline   8. hub fuselage station
        9. posn right of buttline  10. number of blades
       11. blade chord             12. blade radius
       13. lift curve slope        14. rotational velocity
       15. flap mom of inertia     16. delta-3 angle
       17. blade twist

        Vertical Fin
       17. height above waterline  18. fuselage station
       19. posn right of buttline  20. alpha zero lift
       21. CL max                  22. dynamic pressure ratio
       23. lift curve slope        24. Rudder effectiveness

       0. NO CHANGES


       Input the parameter to change:
```

Figure 2.3 Modified Additional Parameters Change Menu Screen 1

The airplane characteristics are where the majority of the modifications and additions come in. JANRAD's stability routine does take into account some wing and fuselage parameters but not enough to sufficiently describe the dynamics of an airplane. Menu screen 1 of the existing JANRAD shows vertical tail parameters significant in a tiltrotor, but it also needs rudder effectiveness (#24) to be added for the airplane mode control derivatives. These two modifications complete those needed on screen #1.

In order to accommodate additional parameters without adding a new screen, the CG information was moved from menu screen 3 to screen 2, as shown in Figure 2.4. This freed up space on menu screen #3 to add a tiltrotor section and some rigging parameters for an airplane. These additional parameters are shown in Figure 2.5. All of these new parameters were also added to the section of STAB.M which inputs the data from scratch.

9

```
          ***    STABILITY AND CONTROL MENU    ***
          *** ADDITIONAL PARAMETERS (2 of 3) ***

  Horizontal Tail
  1. height above waterline      2. fuselage station
  3. posn right of buttline      4. alpha @ zero lift
  5. angle of incidence          6. lift curve slope
  7. dynamic pressure ratio      8. rotor downwash ratio
  9. downwash wrt alpha ratio


  Wing
  10. height above waterline     11. fuselage station
  12. posn right of buttline     13. alpha @ zero lift
  14. angle of incidence         15. lift curve slope
  16. tip cord                   17. root cord
  18. rotor downwash ratio       19. fuselage downwash ratio
  20. flaperon effectiveness


  CG location and Inertias/fuselage parameters
  21. cg ht. above waterline     22. cg fuselage station
  23. cg posn rt of buttline     24. Ixx
  25. Iyy                        26. Izz
  27. Ixz                        28. fuselage downwash ratio


  0. NO CHANGES


  Input the parameter to change:
```

Figure 2.4 Modified Additional Parameters Change Menu Screen 2

```
          ***    STABILITY AND CONTROL MENU    ***
          *** ADDITIONAL PARAMETERS (3 of 3) ***

  NOTAR if available (enter zeros if using tail or tilt rotor)
  1. height above waterline    2. boom fuselage station
  3. boom position left ref    4. NOTAR diameter
  5. swirl angle at boom       6. NOTAR max force
  7. thruster fuselage station


  Tilt Rotor (enter zeros if using tail rotor or NOTAR)
   8. Fuselage CP location       9. Fuselage angle @ zero lift
  10. Fuselage lift slope       11. Fuselage Cmo
  12. Fuselage moment slope     13. Wing aero. center
  14. Wing sweep                15. Wing dihedral
  16. Wing moment coefficient   17. Downwash angle @ zero alpha
  18. Horiz. stab. span eff.    19. Elevator effectiveness


  Rigging
  20. B1 main/in defl (del e)   21. A1 main/in defl (del a)
  22. theta0m/in defl (del c)   23. theta0t/pedal defl (del r or p)
  24. NOTAR sleeve twist/defl   25. max rudder defl
  26. Aileron/in defl (del a)   27. Elevator/in defl (del e)
  28. Rudder/in defl (del r)


  0. NO CHANGES


  Input the parameter to change:
```

Figure 2.5 Modified Additional Parameters Change Menu Screen 3

The final modification for STAB.M is to provide the management software for calling separate routines that perform the tiltrotor analysis. Because the dynamics of the tiltrotor and a helicopter differ so much, the routines that calculate the stability derivatives, HOVER and CRUISE, needed to be modified significantly. These major modifications made it impractical to use the same routines for both a helicopter and a tiltrotor. Therefore, new ones were written for both flight regimes. These are HTLTRGRP.M, to calculate the hover mode stability derivatives themselves, and TLTRHOVR.M, to put the derivatives in the state-space equations/matrices. Similarly, CTLTRGRP.M and TLTRCRUS.M were written for the airplane mode. The development of the equations which make up these routines will be discussed in Chapter III.

The output routines contain the final modifications made to JANRAD. STABOUT.M is the primary output routine for JANRAD's stability section, which calls many other subroutines to perform plotting and other functions. The obvious modification is the addition of the tiltrotor parameters for the input data summary and the airplane type calculated data not present in the helicopter analysis. This routine was written a few years ago and there have been later refinements in MATLAB which make some additional modifications desirable (and in some instances necessary) to run the program with the latest version of MATLAB (version 4.2c). The old STABOUT.M printed the plots to a file using the META command which Matlab's latest version no longer uses. The necessary modification made was in the method of saving and/or printing the many plots generated. STABOUT.M now saves all plots in the Windows Metafile (xxx.wmf) format to be printed using the user's preferred word processor.

One of the key outputs of the JANRAD Stability section is the set of command bandwidth plots. These are simply Bode magnitude plots for all the control inputs to their respective desired outputs. Some important analyses can be made with this data, and with the modifications made to the plot statements these plotting routines should still be a part of the output. However, with the greater interest in time domain analysis in today's stability and control world, frequency domain analysis alone (exemplified by these command bandwidth plots) does not suffice. Incorporating some time domain analysis in this program is relatively painless, especially with Matlab. The structure of the Bode plot routines could be used with the BODE commands being replaced with LSIM commands to give time responses to appropriate control inputs. Most helicopter control inputs produce rate responses in the open-loop case (which this analysis considers). Therefore, simple step inputs would not give favorable responses. The more appropriate responses to be observed would be short time pulses for the

11

longitudinal inputs and doublets for the lateral/directional ones. Doublets should be used because the net control input is zero which keeps the aircraft from entering a turn.

D. TRIMMING ROUTINES

The old JANRAD trimming routine, TRIM, had to be renamed because it conflicted with the Matlab command of the same name. This routine is used by the performance section of JANRAD, as well as the stability and control section. Ref. 9 goes into detail on the methods used to trim the helicopter controls, but these methods did not work well for the tiltrotor. Trimming a tiltrotor in the helicopter mode is somewhat different than trimming a conventional helicopter and is completely different from the trimming the aircraft in the airplane mode. Therefore, new trimming routines for both modes had to be constructed.

First, the helicopter mode trimming routine was modified to accommodate a tiltrotor as well as a conventional helicopter. These changes were primarily designed to remove the tail rotor trimming portion. Therefore, the only trimming parameters that are needed are the collective and the longitudinal cyclic swashplate positions. Longitudinal cyclic is only required for forward flight speeds, so if hover is the only mode which is being analyzed, collective trim is the only parameter to be determined. The old code as well as a routine with the above mentioned modifications (developed apart from this project) were evaluated for the hover mode and the trim results were not acceptably close to the results of Ref. 4. Therefore, the trim values of Ref. 4. were inserted by a script file in order to have comparable trim conditions to start with in determining the stability and control derivatives.

Airplane mode trim is a completely different and much less complicated procedure. In the airplane mode, the flight control positions for straight and level flight are the power (thrust) control and the longitudinal (elevator) control. These positions would simply be determined by balancing out the force and the moment equations. The two equations have the elevator position (δ_e) and wing angle of attack (α_w) as the two unknowns, and they can be determined using a linear technique.

The dominant forces of the tiltrotor used to determine trim in airplane mode are the lift of the three major lifting surfaces. In the following discussion, the standard lift coefficient relationship is used where the lift coefficient, C_L is the lift of the lifting body divided by the dynamic pressure, q and the reference area, A. The lift coefficient is typically a linear relationship with the angle of attack, α and the lift curve slope, C_{L_α} or "a" such that $C_L = C_{L_\alpha}\alpha$.

12

Continuing with these relationships, the following force equations use the standard aerodynamic lift relationship, $L = qAC_L = qAC_{L_\alpha}\alpha = qAa\alpha$ for each lifting surface:

$$\Sigma F = 0 \rightarrow GW = L_{wing} + L_{fuselage} + L_{H\text{-}stab} \tag{2.3}$$

$$L_{wing} = qA_w\left[a_w(\alpha_w - \alpha_{ol}) + \frac{\partial C_{l(flap)}}{\partial \delta_{flap}}\delta_{flap}\right] \tag{2.4}$$

$$L_{H\text{-}stab} = q_H A_H\left(a_H\alpha_H + \frac{\partial C_{l_H}}{\partial \delta_e}\delta_e\right) \tag{2.5}$$

$$\text{where, } \alpha_H = \alpha_w - i_w + i_H - \varepsilon \tag{2.6}$$

$$\text{and, } \varepsilon = \varepsilon_0 + \frac{\partial \varepsilon}{\partial \alpha}\left(\alpha_w - \alpha_{ol_w} + \frac{\partial C_{l(flap)}}{\partial \delta_{flap}}\frac{\delta_{flap}}{a_w}\right) \tag{2.7}$$

$$L_{fuselage} = qA_w C_{l_f} = qA_w\, a_f(\alpha_f - \alpha_{ol_f}) \text{ where} \Rightarrow \alpha_f = \alpha_w - i_w$$

$$= qA_w\, a_f(\alpha_w - i_w - \alpha_{ol_f}) \tag{2.8}$$

These forces must balance out to zero for the aircraft to trim in flight. The resulting force equation, with the unknowns α_w and δ_e accessible, is arranged as follows:

$$\underbrace{q\left[A_w(a_w + a_f) + \frac{q_H}{q}a_H A_H\left(1 - \frac{\partial \varepsilon}{\partial \alpha}\right)\right]}_{\frac{\partial L}{\partial \alpha}}\alpha_w + \underbrace{\left(q\frac{q_H}{q}A_H a_H \frac{\partial C_{l_H}}{\partial \delta_e}\right)}_{\frac{\partial L}{\partial \delta_e}}\delta_e =$$

$$\underbrace{GW + q\left\{A_w[a_w\alpha_{ol_w} + a_f(i_w + \alpha_{ol_f})] - \frac{q_H}{q}A_H a_H(i_H - i_w - \varepsilon_0)\right\}}_{L_{tr}} \tag{2.9}$$

The moments must also balance out. Many of the moments are simply the forces in the force equation creating moments due to their relative distances from the center of gravity. These moments are combined with the aerodynamic moments to form the following moment equations. As with the lift equations, they use the moment coefficient similar to its lift counterpart with the addition of a reference chord length, c to non-dimensionalize the coefficient. The moment coefficient, C_M is defined as the lifting surface moment, M divided (again, as with the lift coefficient) by q and A but also by c. This results in the standard aerodynamic moment relationship, $M = qAcC_M$ for each lifting surface. C_M is also typically a linear relationship with

13

α such that $C_M = C_{M\alpha}\alpha + C_{Mo}$. The following are the remaining equations used to trim the aircraft:

$$\Sigma M = 0 \rightarrow M_{wing} + M_{fuselage} + M_{H\text{-}stab} = 0 \tag{2.10}$$

$$M_{wing} = qA_wc_w(C_{mo_w} + C_{m\alpha_w}\alpha_w) - l_w L_w \tag{2.11}$$

$$M_{H\text{-}stab} = -l_H L_H \tag{2.12}$$

$$M_{fuselage} = qA_wc_w (C_{mo_f} + C_{m\alpha_f}\alpha_f) \tag{2.13}$$

$$\Sigma M = qA_wc_w(C_{mo_w} + C_{mo_f} + C_{m\alpha_f}\alpha_f) - l_w L_w - l_H L_H - l_f L_f \tag{2.14}$$

Substituting in for the lift terms and rearranging so the two unknowns are accessible leads to:

$$\underbrace{qA_w[c_wC_{m\alpha_f} - l_w a_w - l_f a_f - l_H \tfrac{q_H}{q} a_H \tfrac{A_H}{A_w}(1 - \tfrac{\partial\varepsilon}{\partial\alpha})]}_{\frac{\partial M}{\partial\alpha_w}} \alpha_w - \underbrace{(q\tfrac{q_H}{q}A_H a_H l_H \tfrac{\partial C_{l_H}}{\partial\delta_e})}_{\frac{\partial M}{\partial\delta_e}} \delta_e = \tag{2.15}$$

$$\underbrace{q\{A_w[c_w (C_{m\alpha_f} i_w - C_{mo_w} - C_{mo_f}) - l_w a_w \alpha_{ol_w}] - l_f a_f (i_w + \alpha_{ol_f}) + l_H \tfrac{q_H}{q}A_H a_H (i_H - i_w - \varepsilon_o)\}}_{M_{tr}}$$

Combining the force and the moment equations gives the following matrix equation in α_w and δ_e.

$$\begin{Bmatrix} \dfrac{\partial L}{\partial\alpha_w}\alpha_w + \dfrac{\partial L}{\partial\delta_e}\delta_e = L_{tr} \\[2mm] \dfrac{\partial M}{\partial\alpha_w}\alpha_w + \dfrac{\partial M}{\partial\delta_e}\delta_e = M_{tr} \end{Bmatrix} \Rightarrow \begin{bmatrix} \dfrac{\partial L}{\partial\alpha_w} & \dfrac{\partial L}{\partial\delta_e} \\[2mm] \dfrac{\partial M}{\partial\alpha_w} & \dfrac{\partial M}{\partial\delta_e} \end{bmatrix} \begin{bmatrix} \alpha_w \\ \delta_e \end{bmatrix} = \begin{bmatrix} L_{tr} \\ M_{tr} \end{bmatrix} \tag{2.16}$$

All of these force and moment equations are incorporated in the routine, APTRIM.M listed in Appendix M.

14

III. DEVELOPMENT OF STABILITY DERIVATIVES

A.   HELICOPTER MODE

The focus of this project is the development of the stability derivatives that make up the state space model for the aircraft. Before this development can be discussed, the underlying assumptions must be highlighted. Since this project is a modification to JANRAD (helicopter) routines, the assumptions used in its original development are listed below.

1. The aircraft is a rigid body (including the rotor blades).
2. Small climb angles, pitch attitudes, and angles of attack.
3. Linearity in all derivatives and partial derivatives (first order Taylor Series approximation)
4. Constant or average lift curve slope for the proprotor blades.
5. Uniform inflow through rotor system.
6. Aircraft out of ground effect.

Additional assumptions were also made for the tiltrotor with respect to its helicopter mode analysis. These assumptions are summarized below.

1. The two proprotors behave like two counter-rotating helicopter rotor heads.
2. Right proprotor rotates CCW (as viewed from the top) and left, CW.
3. Airframe out of rotor wake influence.

The approach this analysis takes is to use the equations in Ref. 6 and Ref. 7 where they apply and modify them where they do not. The first set of derivatives are the basic hover derivatives. As discussed in References 6 and 7, these are subsequently used in the stability derivatives themselves. Some of these derivatives have opposite signs if the rotational direction of the rotor system is CW. Therefore, a careful study of the sign convention of each term was made. So that the derivatives for each proprotor head have a consistent sign, the convention for the flapping angles are such that $a_{1s\,(\text{left})} = a_{1s\,(\text{right})}$ and $b_{1s\,(\text{left})} = -b_{1s\,(\text{right})}$. With this convention, the basic rotor derivatives in Ref. 6 can be used. These derivatives are depicted in Table 3.1.

15

Another set of convention used in this analysis is one for the lateral/directional inputs. The JANRAD helicopter section (along with Ref. 6) uses the notation of A_1 (or $\frac{\partial()}{\partial A_1}$ for its lateral derivatives) where A_1 is the lateral swashplate angle. However, as discussed in Chapter II, roll inputs are made with a tiltrotor by having differential collective on each rotor. For consistency, the tiltrotor lateral derivatives will use the same notation as the helicopter section for this analysis even though there is no true lateral feathering, A_1 with a tiltrotor. Similarly for directional inputs, the notation, θ_{oT} (as in $\frac{\partial()}{\partial\theta_{oT}}$) is used in the code but for clarity, $\frac{\partial()}{\partial\delta_p}$ and $\frac{\partial()}{\partial\Delta B_1}$ are used interchangeably in the discussion because they are more descriptive for a tiltrotor directional pedal input.

Basic Rotor Derivatives

$$\frac{\partial\mu}{\partial\dot{x}}, \frac{\partial\lambda'}{\partial\dot{z}} = \frac{1}{\Omega R}$$

$$\frac{\partial a_{1s}}{\partial\mu} = \tfrac{8}{3}\theta_o + 2\theta_1 - 2\frac{v_1}{\Omega R}$$

$$\frac{\partial b_{1s}}{\partial\mu} = \tfrac{4}{3}a_o$$

$$\left.\begin{array}{c} \dfrac{\partial C_T/\sigma}{\partial\theta_o} \\[2em] \dfrac{\partial C_Q/\sigma}{\partial\theta_o} \end{array}\right\} \quad \text{Calculated numerically in the Performance Section}$$

$$\frac{\partial C_T/\sigma}{\partial\lambda'} = \left(\frac{8}{a} + \frac{\sqrt{\sigma/2}}{\sqrt{C_T/\sigma}}\right)^{-1}$$

$$\frac{\partial C_Q/\sigma}{\partial\lambda'} = -\frac{a}{4}\left(\theta_{.75} - 2\frac{v_1}{\Omega R}\right)$$

$$\left(\frac{\partial a_{1s}}{\partial p}\right)_R = -\left(\frac{\partial a_{1s}}{\partial p}\right)_L, \ -\left(\frac{\partial b_{1s}}{\partial q}\right)_R = \left(\frac{\partial b_{1s}}{\partial q}\right)_L = \frac{1}{\Omega}\left[1 - \frac{192\left(\frac{e}{R}\right)}{\gamma^2\left(1-\frac{e}{R}\right)^5}\right]$$

$$\left(\frac{\partial a_{1s}}{\partial q}, \frac{\partial b_{1s}}{\partial p}\right)_R = \left(\frac{\partial a_{1s}}{\partial q}, -\frac{\partial b_{1s}}{\partial p}\right)_L = -\frac{16}{\gamma\Omega\left(1-\frac{e}{R}\right)^2} - \frac{12\left(\frac{e}{R}\right)}{\gamma\Omega\left(1-\frac{e}{R}\right)^3}$$

Continued

Table 3.1

$$\left(\frac{\partial a_{1s}}{\partial A_1},\frac{\partial b_{1s}}{\partial B_1}\right)_R = \left(\frac{\partial a_{1s}}{\partial A_1},\frac{\partial b_{1s}}{\partial B_1}\right)_L = \frac{12\left(\frac{e}{R}\right)}{\gamma\left(1-\frac{e}{R}\right)^3}$$

$$\left(\frac{\partial a_{1s}}{\partial B_1},-\frac{\partial b_{1s}}{\partial A_1}\right)_R = \left(\frac{\partial a_{1s}}{\partial B_1},-\frac{\partial b_{1s}}{\partial A_1}\right)_L = -\left(1+\frac{144\left(\frac{e}{R}\right)^2}{\gamma^2\left(1-\frac{e}{R}\right)^6}\right)^{-1}$$

$$\left(\frac{\partial C_H/\sigma}{\partial a_{1s}},\frac{\partial C_Y/\sigma}{\partial b_{1s}}\right)_R = \left(\frac{\partial C_H/\sigma}{\partial a_{1s}},\frac{\partial C_Y/\sigma}{\partial b_{1s}}\right)_L = \frac{3}{2}(C_T/\sigma)\left(1-\frac{a}{18}\frac{\theta_{.75}}{C_T/\sigma}\right)$$

$$\left(\frac{\partial M}{\partial a_{1s}},\frac{\partial R}{\partial b_{1s}}\right)_R = \left(\frac{\partial M}{\partial a_{1s}},\frac{\partial R}{\partial b_{1s}}\right)_L = \frac{\frac{3}{4}\frac{e}{R}A_b\rho R(\Omega R)^2 a}{\gamma}$$

Table 3.1 (continued)

1. Force Derivatives

Due to the counter rotating effect, many derivatives are the same for each proprotor head, and others are equal but opposite in sign. With the force derivatives, this phenomena causes the derivatives with the same sign to be additive and the ones of opposite sign to cancel each other.

a. *X-Force Derivatives*

An example of the additive effect is the X-force perturbation with respect to a small perturbation in forward velocity, or $\left(\frac{\partial X}{\partial \dot{x}}\right)_{total}$, is simply twice the effect of each rotor alone. The negating effect is demonstrated by the in sideward velocity derivative, $\left(\frac{\partial X}{\partial \dot{y}}\right)_{total}$. Here, due to the proprotors rotating in opposite directions, the right derivative is equal in magnitude to the one for the left side but of opposite sign. Therefore, the resulting force derivative is zero, as shown in Table 3.2.

X-force Derivatives

<u>Development</u>   <u>(=>) Derivative</u>

$$\left(\frac{\partial X}{\partial \dot{x}}\right)_{right} = \left(\frac{\partial X}{\partial \dot{x}}\right)_{left} = -\rho A_b (\Omega R)^2 \left(\frac{\partial C_H/\sigma}{\partial a_{1s}}\right)\left(\frac{\partial a_{1s}}{\partial \mu}\right)\left(\frac{\partial \mu}{\partial \dot{x}}\right)$$

$$\Rightarrow \left(\frac{\partial X}{\partial \dot{x}}\right)_{total} = -2\rho A_b (\Omega R)^2 \left(\frac{\partial C_H/\sigma}{\partial a_{1s}}\right)\left(\frac{\partial a_{1s}}{\partial \mu}\right)\left(\frac{\partial \mu}{\partial \dot{x}}\right)$$

$$\left(\frac{\partial X}{\partial \dot{y}}\right)_{right} = -\left(\frac{\partial X}{\partial \dot{y}}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial X}{\partial \dot{y}}\right)_{total} = 0$$

$$\left(\frac{\partial X}{\partial p}\right)_{right} = -\left(\frac{\partial X}{\partial p}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial X}{\partial p}\right)_{total} = 0$$

$$\left(\frac{\partial X}{\partial q}\right)_{right} = -\left(\frac{\partial X}{\partial q}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial X}{\partial q}\right)_{total} = 0$$

<div align="center">*** <u>Control Inputs</u> ***</div>

$$\left(\frac{\partial X}{\partial \theta_o}\right)_{right} = \left(\frac{\partial X}{\partial \theta_o}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial X}{\partial \theta_o}\right)_{total} = -2\rho A_b (\Omega R)^2 (\bar{a}_{1s} + i_m)\left(\frac{\partial C_T/\sigma}{\partial \theta_o}\right)$$

$$\left(\frac{\partial X}{\partial B_1}\right)_{right} = \left(\frac{\partial X}{\partial B_1}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial X}{\partial B_1}\right)_{total} = -2\rho A_b (\Omega R)^2 \left(\frac{\partial C_H/\sigma}{\partial a_{1s}}\right)\left(\frac{\partial a_{1s}}{\partial B_1}\right)$$

$$\left(\frac{\partial X}{\partial A_1}\right)_{right} = -\left(\frac{\partial X}{\partial \theta_o}\right)_{right} \quad \& \quad \left(\frac{\partial X}{\partial A_1}\right)_{left} = \left(\frac{\partial X}{\partial \theta_o}\right)_{left}$$

$$\left(\frac{\partial X}{\partial A_1}\right)_{right} = -\left(\frac{\partial X}{\partial A_1}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial X}{\partial A_1}\right)_{total} = 0$$

<div align="center">**Table 3.2**</div>

b.   *Y- Force Derivatives*

The lateral derivatives are a bit more interesting and challenging. For the same reasons as with the X-force derivatives, many of them turn out to be zero also. However, due to

<div align="center">18</div>

the significant lateral distance (y_m) between the rotors and the CG, the forces due to yaw perturbations are significant, unlike a conventional helicopter. The yaw perturbation itself has negligible effect, but the forward velocity due to the yaw and the y_m distance is where the effect takes place. This effect would seem to be a negating one, but the counter rotation results in the relation, $\left(\dfrac{\partial Y}{\partial \dot{x}}\right)_{right} = -\left(\dfrac{\partial Y}{\partial \dot{x}}\right)_{left}$. This causes this effect to be additive as in the $\left(\dfrac{\partial Y}{\partial r}\right)_{total}$ derivative shown in Table 3.3. The control derivatives show the same effect, where the lateral inputs are additive and the longitudinal inputs cancel each other.

<div align="center">

Y-force Derivatives

</div>

Development   (=>) Derivative

$$\left(\frac{\partial Y}{\partial \dot{x}}\right)_{right} = -\left(\frac{\partial Y}{\partial \dot{x}}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial Y}{\partial \dot{x}}\right)_{total} = 0$$

$$\left(\frac{\partial Y}{\partial \dot{y}}\right)_{right} = -\left(\frac{\partial Y}{\partial \dot{y}}\right)_{left} = 0 \qquad\qquad \Rightarrow \left(\frac{\partial Y}{\partial \dot{y}}\right)_{total} = 0$$

$$\left(\frac{\partial Y}{\partial p}\right)_{right} = -\left(\frac{\partial Y}{\partial p}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial Y}{\partial p}\right)_{total} = 0$$

$$\left(\frac{\partial Y}{\partial q}\right)_{right} = -\left(\frac{\partial Y}{\partial q}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial Y}{\partial q}\right)_{total} = 0$$

$$\left(\frac{\partial Y}{\partial r}\right)_{right} = -\left(\frac{\partial Y}{\partial \dot{x}}\right)_{right} y_m \;\&\; \left(\frac{\partial Y}{\partial r}\right)_{left} = \left(\frac{\partial Y}{\partial \dot{x}}\right)_{left} y_m$$

$$\Rightarrow \left(\frac{\partial Y}{\partial r}\right)_{total} = -2\rho A_b (\Omega R)^2 \left(\frac{\partial C_Y/\sigma}{\partial b_{1s}}\right) \left(\frac{\partial b_{1s}}{\partial \mu}\right) \left(\frac{\partial \mu}{\partial \dot{x}}\right) y_m$$

<div align="center">

*** Control Inputs ***

</div>

$$\left(\frac{\partial Y}{\partial B_1}\right)_{right} = -\left(\frac{\partial Y}{\partial B_1}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial Y}{\partial B_1}\right)_{total} = 0$$

$$\left(\frac{\partial Y}{\partial \theta_o}\right)_{right} = -\left(\frac{\partial Y}{\partial \theta_o}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial Y}{\partial \theta_o}\right)_{total} = 0$$

Continued

<div align="center">

Table 3.3

</div>

Development   (=>) Derivative

$$\left(\frac{\partial Y}{\partial A_1}\right)_{right} = -\left(\frac{\partial Y}{\partial \theta_o}\right)_{right} \quad \&$$

$$\left(\frac{\partial Y}{\partial A_1}\right)_{left} = \left(\frac{\partial Y}{\partial \theta_o}\right)_{left} = -\left(\frac{\partial Y}{\partial \theta_o}\right)_{right}$$

$$\left(\frac{\partial Y}{\partial A_1}\right)_{total} = -\left(\frac{\partial Y}{\partial \theta_o}\right)_{right} + \left(\frac{\partial Y}{\partial \theta_o}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial Y}{\partial A_1}\right)_{total} = -2\rho A_b(\Omega R)^2\, \bar{b}_{1s}\left(\frac{\partial C_T/\sigma}{\partial \theta_o}\right)$$

$$\left(\frac{\partial Y}{\partial \delta_p}\right)_{right} = -\left(\frac{\partial Y}{\partial B_1}\right)_{right} \quad \&$$

$$\left(\frac{\partial Y}{\partial \delta_p}\right)_{left} = \left(\frac{\partial Y}{\partial B_1}\right)_{left} = -\left(\frac{\partial Y}{\partial B_1}\right)_{right}$$

$$\left(\frac{\partial Y}{\partial \delta_p}\right)_{total} = \left(\frac{\partial Y}{\partial B_1}\right)_{right} + \left(\frac{\partial Y}{\partial B_1}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial Y}{\partial \delta_p}\right)_{total} = -2\left(\frac{\partial Y}{\partial B_1}\right)_{right} = -2\rho A_b(\Omega R)^2\left(\frac{\partial C_Y/\sigma}{\partial b_{1s}}\right)\left(\frac{\partial b_{1s}}{\partial B_1}\right)$$

Table 3.3 (continued)

c.   *Z-Force Derivatives*

The Z-force derivatives for the longitudinal cases are identical to the helicopter ones in this analysis. They are additive and the equations used in Ref(s). 6 and 7 are applicable by simply multiplying them by two. The lateral derivatives intuitively cancel each other, and the results are shown in Table 3.4.

20

Z-force Derivatives

Development   (=>) Derivative

$$\left(\frac{\partial Z}{\partial \dot{x}}\right)_{right} = \left(\frac{\partial Z}{\partial \dot{x}}\right)_{left} = 0 \qquad\qquad \Rightarrow \left(\frac{\partial Z}{\partial \dot{x}}\right)_{total} = 0$$

$$\left(\frac{\partial Z}{\partial \dot{z}}\right)_{right} = \left(\frac{\partial Z}{\partial \dot{z}}\right)_{left} = -\rho A_b(\Omega R)^2 \left(\frac{\partial C_T/\sigma}{\partial \lambda'}\right)\left(\frac{\partial \lambda'}{\partial \dot{z}}\right) \cdots$$

$$\Rightarrow \left(\frac{\partial Z}{\partial \dot{z}}\right)_{total} = -2\rho A_b(\Omega R)^2 \left(\frac{\partial C_T/\sigma}{\partial \lambda'}\right)\left(\frac{\partial \lambda'}{\partial \dot{z}}\right)$$

$$\left(\frac{\partial Z}{\partial \theta_o}\right)_{right} = \left(\frac{\partial Z}{\partial \theta_o}\right)_{left} = -\rho A_b(\Omega R)^2 \left(\frac{\partial C_T/\sigma}{\partial \theta_o}\right)$$

$$\Rightarrow \left(\frac{\partial Z}{\partial \theta_o}\right)_{total} = -2\rho A_b(\Omega R)^2 \left(\frac{\partial C_T/\sigma}{\partial \theta_o}\right)$$

Table 3.4

2.   Moment Derivatives

The moment derivatives have forces embedded in them, along with appropriate moment arms, l_m, y_m, and h_m. The forces of opposite sign here, cause the moments associated with them to be addditive. Conversely, the forces of the same sign, have moments that cancel each other. Again, each axis will be presented separately.

a.   *Roll Moment, R Derivatives*

Many of the roll dervatives, including the lateral ones, fall out because of the counter rotation effect. The ones that do not are the roll disturbance and the roll/yaw coupling derivatives. For the roll disturbance derivative, $\frac{\partial R}{\partial p}$, there are two primary effects to consider. First, is the roll effect of each rotor due to a roll disturbance, $\left(\frac{\partial R}{\partial p}\right)_{R\ or\ L} = \frac{\partial R}{\partial b_{1s}}\frac{\partial b_{1s}}{\partial p}$. When the contributions for each rotor are combined, however, this pure roll effect cancels. Second, is the moment induced by the Z-force along with the lateral moment arm, y_m depicted by: $\left(\frac{\partial Z}{\partial p}\right)_{R\ or\ L} = \left(\frac{\partial Z}{\partial \dot{z}}\right)\left(\frac{\partial \dot{z}}{\partial p}\right)$ and is simplified by knowing that $\left(\frac{\partial \dot{z}}{\partial p}\right)_{(R\ or\ L)}$ is merely equal to y_m (in magnitude). The same analysis was applied to all the other roll moment derivatives and is summarized in Table 3.5.

21

Roll Moment, R Derivatives

$$\left(\frac{\partial R}{\partial \dot{x}}\right)_{\text{right}} = -\left(\frac{\partial R}{\partial \dot{x}}\right)_{\text{left}} \qquad \Rightarrow \left(\frac{\partial R}{\partial \dot{x}}\right)_{\text{total}} = 0$$

$$\left(\frac{\partial R}{\partial \dot{y}}\right)_{\text{right}} = -\left(\frac{\partial M}{\partial \dot{x}}\right)_{\text{right}} = -\frac{\partial M}{\partial a_{1s}}\frac{\partial a_{1s}}{\partial \mu}\frac{\partial \mu}{\partial \dot{x}} + \left(\frac{\partial X}{\partial \dot{x}}\right)_{\text{right}} h_m$$

$$\Rightarrow \left(\frac{\partial R}{\partial \dot{y}}\right)_{\text{total}} = \left(\frac{\partial Y}{\partial \dot{y}}\right)_{\text{total}} h_m - 2\frac{\partial M}{\partial a_{1s}}\frac{\partial a_{1s}}{\partial \mu}\frac{\partial \mu}{\partial \dot{x}}$$

$$\left(\frac{\partial R}{\partial \dot{z}}\right)_{\text{right}} = -\left(\frac{\partial R}{\partial \dot{z}}\right)_{\text{left}} \qquad \Rightarrow \left(\frac{\partial R}{\partial \dot{z}}\right)_{\text{total}} = 0$$

$$\left(\frac{\partial R}{\partial p}\right)_{\text{right}} = \left(\frac{\partial R}{\partial b_{1s}}\right)_R \left(\frac{\partial b_{1s}}{\partial p}\right)_R + \left(\frac{\partial Z}{\partial p}\right)_R y_m$$

$$= \left(\frac{\partial R}{\partial b_{1s}}\right)_R \left(\frac{\partial b_{1s}}{\partial p}\right)_R + \left(\frac{\partial Z}{\partial \dot{z}}\right)\left(\frac{\partial \dot{z}}{\partial p}\right) y_m$$

$$= \left(\frac{\partial R}{\partial b_{1s}}\right)_R \left(\frac{\partial b_{1s}}{\partial p}\right)_R + \left(\frac{\partial Z}{\partial \dot{z}}\right) y_m^2$$

$$\left(\frac{\partial R}{\partial p}\right)_{\text{left}} = \left(\frac{\partial R}{\partial b_{1s}}\right)_L \left(\frac{\partial b_{1s}}{\partial p}\right)_L + \left(\frac{\partial Z}{\partial \dot{z}}\right)\left(\frac{\partial \dot{z}}{\partial p}\right) y_m$$

$$= -\left(\frac{\partial R}{\partial b_{1s}}\right)_R \left(\frac{\partial b_{1s}}{\partial p}\right)_R + \left(\frac{\partial Z}{\partial \dot{z}}\right) y_m^2$$

$$\left(\frac{\partial R}{\partial p}\right)_{\text{total}} = \left(\frac{\partial R}{\partial p}\right)_R + \left(\frac{\partial R}{\partial p}\right)_L \qquad \Rightarrow \left(\frac{\partial R}{\partial p}\right)_{\text{total}} = \left(\frac{\partial Z}{\partial \dot{z}}\right)_{\text{total}} y_m^2$$

$$\left(\frac{\partial R}{\partial q}\right)_{\text{right}} = -\left(\frac{\partial R}{\partial q}\right)_{\text{left}} \qquad \Rightarrow \left(\frac{\partial R}{\partial q}\right)_{\text{total}} = 0$$

$$\left(\frac{\partial R}{\partial r}\right)_{\text{right}} = \left(\frac{\partial R}{\partial \dot{x}}\right)_R y_m$$

$$= \left[\left(\frac{\partial R}{\partial b_{1s}}\right)_R \frac{\partial b_{1s}}{\partial \mu}\frac{\partial \mu}{\partial \dot{x}} + \left(\frac{\partial Y}{\partial \dot{x}}\right)_R h_m\right] y_m$$

Continued

Table 3.5

Development   (=>) Derivative

$$\left(\frac{\partial R}{\partial r}\right)_{\text{left}} = -\left(\frac{\partial R}{\partial \dot{x}}\right)_L y_{\text{m}}$$

$$= -\left[\left(\frac{\partial R}{\partial b_{1s}}\right)_L \frac{\partial b_{1s}}{\partial \mu}\frac{\partial \mu}{\partial \dot{x}} + \left(\frac{\partial Y}{\partial \dot{x}}\right)_L h_{\text{m}}\right] y_{\text{m}}$$

$$\left(\frac{\partial R}{\partial r}\right)_{\text{total}} = \left(\frac{\partial R}{\partial r}\right)_{\text{right}} + \left(\frac{\partial R}{\partial r}\right)_{\text{left}} = 2\left(\frac{\partial Y}{\partial \dot{x}}\right)_R h_{\text{m}}\, y_{\text{m}}$$

$$\Rightarrow \left(\frac{\partial R}{\partial r}\right)_{\text{total}} = 2\rho A_b (\Omega R)^2 \left(\frac{\partial C_Y/\sigma}{\partial b_{1s}}\right)\left(\frac{\partial b_{1s}}{\partial \mu}\right)\left(\frac{\partial \mu}{\partial \dot{x}}\right) h_{\text{m}}\, y_{\text{m}}$$

** Control Inputs **

$$\left(\frac{\partial R}{\partial \delta_a}\right)_{\text{right}} = \left(\frac{\partial R}{\partial A_1}\right)_{\text{right}} = \left(\frac{\partial Y}{\partial \theta_o}\right)_R h_{\text{m}} - \left(\frac{\partial Z}{\partial \theta_o}\right)_R y_{\text{m}}$$

$$= \left(\frac{\partial R}{\partial \delta_a}\right)_{\text{left}}$$

$$\Rightarrow \left(\frac{\partial R}{\partial \delta_a}\right)_{\text{total}} = \left(\frac{\partial Y}{\partial A_1}\right)_{\text{total}} h_{\text{m}} - \left(\frac{\partial Z}{\partial \theta_o}\right)_{\text{total}} y_{\text{m}}$$

$$\left(\frac{\partial R}{\partial \delta_p}\right)_{\text{total}} = \left(\frac{\partial R}{\partial \Delta B_1}\right)_{\text{total}} = \left(\frac{\partial R}{\partial B_1}\right)_R + \left(\frac{\partial R}{\partial B_1}\right)_L$$

$$\Rightarrow \left(\frac{\partial R}{\partial \delta_p}\right)_{\text{total}} = 2\left(\frac{\partial R}{\partial b_{1s}}\frac{\partial b_{1s}}{\partial B_1} + \frac{\partial Y}{\partial b_{1s}}h_{\text{m}} + \frac{\partial Z}{\partial B_1}y_{\text{m}}\right)$$

Table 3.5 (continued)

b.   *Pitch Moment, M Derivatives*

Due to tiltrotor aircraft being symmetric along the longitudinal axis, pitch moments are only generated by longitudinal forces and moments. An example of a pitch moment derivative that is non-zero is the $\frac{\partial M}{\partial \dot{x}}$ derivative. This moment is not effected by the direction of rotation, therefore the left and the right rotor effects will be additive, resulting in the relation shown in Table 3.6, which is simply twice that found in Ref. 6. The lateral perturbations, such as the $\frac{\partial M}{\partial \dot{y}}$ derivative, cancel due to the counter rotation effect leaving all the lateral perturbation derivatives equal to zero. The collective control derivative, $\frac{\partial M}{\partial \theta_o}$ has

contributions from the X-force with the vertical moment arm, y_m and the Z-force coupled with the longitudinal distance, l_m to produce the result also shown in Table 3.6.

Pitch Moment, M Derivatives

<u>Development</u>   <u>(=>) Derivative</u>

$$\left(\frac{\partial M}{\partial \dot{x}}\right)_{right} = \left(\frac{\partial M}{\partial \dot{x}}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial \dot{x}}\right)_{total} = 2\frac{\partial M}{\partial a_{1s}}\frac{\partial a_{1s}}{\partial \mu}\frac{\partial \mu}{\partial \dot{x}} - \left(\frac{\partial X}{\partial \dot{x}}\right)_{total} h_m$$

$$\left(\frac{\partial M}{\partial \dot{y}}\right)_{right} = \left(\frac{\partial R}{\partial \dot{x}}\right)_{right} = -\left(\frac{\partial M}{\partial \dot{y}}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial \dot{y}}\right)_{total} = 0$$

$$\left(\frac{\partial M}{\partial \dot{z}}\right)_{right} = \left(\frac{\partial M}{\partial \dot{z}}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial \dot{z}}\right)_{total} = -\left(\frac{\partial X}{\partial \dot{z}}\right)_{total} h_m + \left(\frac{\partial Z}{\partial \dot{z}}\right)_{total} l_m$$

$$\left(\frac{\partial M}{\partial p}\right)_{right} = -\left(\frac{\partial M}{\partial p}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial p}\right)_{total} = 0$$

$$\left(\frac{\partial M}{\partial q}\right)_{right} = \frac{\partial M}{\partial a_{1s}}\frac{\partial a_{1s}}{\partial q} - \left(\frac{\partial X}{\partial q}\right)_{right} h_m$$
$$= \left(\frac{\partial M}{\partial q}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial q}\right)_{total} = -\left(\frac{\partial X}{\partial q}\right)_{total} h_m$$

$$\left(\frac{\partial M}{\partial r}\right)_{right} = -\left(\frac{\partial X}{\partial r}\right)_{right} h_m + \left(\frac{\partial Z}{\partial r}\right)_{right} l_m$$
$$= -\left(\frac{\partial M}{\partial r}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial r}\right)_{total} = 0$$

** Control Inputs ***

$$\left(\frac{\partial M}{\partial B_1}\right)_{right} = \left(\frac{\partial M}{\partial B_1}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial B_1}\right)_{total} = 2\frac{\partial M}{\partial a_{1s}}\frac{\partial a_{1s}}{\partial B_1} - \frac{\partial X}{\partial B_1} h_m$$

Continued

Table 3.6

24

Pitch Moment, M Derivatives (continued)

<u>Development</u>   <u>(=>) Derivative</u>

$$\left(\frac{\partial M}{\partial \theta_o}\right)_{right} = \left(\frac{\partial M}{\partial \theta_o}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial \theta_o}\right)_{total} = -\left(\frac{\partial X}{\partial \theta_o}\right)_{total} h_m + \left(\frac{\partial Z}{\partial \theta_o}\right)_{total} l_m$$

$$\left(\frac{\partial M}{\partial A_1}\right)_{right} = -\left(\frac{\partial M}{\partial A_1}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial M}{\partial A_1}\right)_{total} = 0$$

Table 3.6 (continued)

c.   *Yaw Moment, N Derivatives*

Yaw moments are, as are the roll moments, lateral mode derivatives. These also result in all the longitudinal disturbances having no contribution to yaw moments. The lateral disturbances cause no pure yaw moments, yet there are significant moments generated by the lateral forces due to the rotor distances from the aircraft cg. These couplings are the X-forces with the y_m distance and the Y-force with the l_m distance. The two yaw moment derivatives due to the lateral rates are somewhat more complicated due to the moments being generated by forces which are themselves generated by the angular rates. This effect was discussed in the roll moment section where the relation, $\left|\left(\frac{\partial \dot{z}}{\partial p}\right)_{(R \, or \, L)}\right| = y_m$, was used. Using the same analysis, $\left|\left(\frac{\partial \dot{x}}{\partial r}\right)_{(R \, or \, L)}\right| = y_m$ and both of these relations were used here, to simplify the roll and yaw perturbation derivatives shown in Table 3.7.

The control derivatives are also quite nontrivial. The yaw moment due to a lateral cyclic input, $\frac{\partial N}{\partial A_1}$ has components due to the X-force with the vertical distance, h_m and Y-force with the longitudinal distance, l_m. The yaw moment due to directional pedal input, (by convention) $\frac{\partial N}{\partial \Delta B_1}$ has the same coupling effects as the lateral input, as depicted in the last two derivatives in Table 3.7.

Yaw Moment, N Derivatives

$$\left(\frac{\partial N}{\partial \dot{x}}\right)_{right} = -\left(\frac{\partial N}{\partial \dot{x}}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial N}{\partial \dot{x}}\right)_{total} = 0$$

$$\left(\frac{\partial N}{\partial \dot{y}}\right)_{right} = \left(\frac{\partial N}{\partial \dot{y}}\right)_{left}$$

$$= -\left(\frac{\partial X}{\partial \dot{y}}\right)_{right} y_m - \left(\frac{\partial Y}{\partial \dot{y}}\right)_{right} l_m$$

$$\Rightarrow \left(\frac{\partial N}{\partial \dot{y}}\right)_{total} = 2\rho A_b(\Omega R)^2 \left(\frac{\partial C_y/\sigma}{\partial b_{1s}}\right)\left(\frac{\partial b_{1s}}{\partial \mu}\right)\left(\frac{\partial \mu}{\partial \dot{x}}\right) y_m - \left(\frac{\partial Y}{\partial \dot{y}}\right)_{total} l_m$$

$$\left(\frac{\partial N}{\partial \dot{z}}\right)_{right} = -\left(\frac{\partial N}{\partial \dot{z}}\right)_{left} \qquad\qquad \Rightarrow \left(\frac{\partial N}{\partial \dot{z}}\right)_{total} = 0$$

$$\Rightarrow \left(\frac{\partial N}{\partial p}\right)_{total} = \left[2\rho A_b(\Omega R)^2 \left(\frac{\partial C_H/\sigma}{\partial a_{1s}}\right)\left(\frac{\partial a_{1s}}{\partial p}\right) - \frac{\partial X}{\partial \dot{z}} y_m\right] y_m - \left(\frac{\partial Y}{\partial p}\right)_{total} l_m$$

$$\Rightarrow \left(\frac{\partial N}{\partial r}\right)_{total} = \left[-\left(\frac{\partial Y}{\partial r}\right)_{total} l_m + \left(\frac{\partial X}{\partial \dot{x}}\right)_{total} y_m\right] y_m$$

** Control Inputs **

$$\left(\frac{\partial N}{\partial \delta_a}\right)_{right} = \left(\frac{\partial N}{\partial A_1}\right)_{right} = \left(\frac{\partial X}{\partial \theta_o}\right)_R y_m - \left(\frac{\partial Y}{\partial \delta_a}\right)_R l_m$$

$$= \left(\frac{\partial N}{\partial A_1}\right)_{left}$$

$$\Rightarrow \left(\frac{\partial N}{\partial \delta_a}\right)_{total} = \left(\frac{\partial X}{\partial \theta_o}\right)_{total} y_m - \left(\frac{\partial Y}{\partial \delta_a}\right)_{total} l_m$$

$$\left(\frac{\partial N}{\partial \Delta B_1}\right)_{total} = \left(\frac{\partial N}{\partial B_1}\right)_R + \left(\frac{\partial N}{\partial B_1}\right)_L$$

$$\Rightarrow \left(\frac{\partial N}{\partial \Delta B_1}\right)_{total} = \left(\frac{\partial X}{\partial B_1}\right)_{total} y_m + \left(\frac{\partial Y}{\partial B_1}\right)_{total} l_m$$

Table 3.7

B. AIRPLANE MODE

As with the helicopter mode, there are some underlying assumptions that need to be made for our simplified model of a tiltrotor flying in airplane mode. The same assumptions from the helicopter section are made with the following additional ones:

1. The wing has constant airfoil section, (i.e., constant a_w)
2. Rotor effects on the airflow over the wing and other parts of the aircraft are negligible.

JANRAD helicopter analysis takes into account airplane type features such as wings, horizontal and vertical stabilizers as discussed in Chapter II. This analysis will use these previously developed relations where they seem to be effective, but some will be modified or re-derived as necessary.

All derivatives are based on a trimmed flight condition and the resulting forces that balance out for each of the three axes. As discussed in the trim section, the forces which are determined by the trim routine are in the wind axis. These forces are lift ($L_{()}$) and drag ($D_{()}$) and are determined for all the aerodynamic components (the fuselage, wing, horizontal stabilizer and vertical tail). These forces are transformed into the body axis by the relations shown at the beginning of each component sections to follow. Derivatives for each component are then calculated separately and subsequently added to arrive at the total aircraft derivatives.

The flight control derivatives are where Ref. 6 and Ref. 7 fall short for a tiltrotor. Though JANRAD handles many airplane aerodynamic qualities (for the compound helicopter analysis), flight controls are not included because compound helicopters are controlled by normal, helicopter (swashplate) controls. For this reason, the airplane mode flight control derivatives were derived or developed using relations in Ref. 5.

1. Fuselage Derivatives

The first few of the nondimensional derivatives (to be used in the dimensional derivatives later) are taken from Ref. 6 with the rotor downwash effect removed since there is no main rotor above the fuselage affecting the airflow around it. The remaining nondimensional fuselage derivatives of Ref. 6 are determined from performance and aerodynamic charts. This is not practical for an interactive preliminary design tool like JANRAD. These values were

27

approximated from the charts and "hard wired" into the JANRAD helicopter routine, but are not as useful for the more slender fuselage shape a tiltrotor has. For this reason, these nondimensional fuselage derivatives were taken from XV-15 wind tunnel data [Ref. 2].

The dimensional derivatives of Ref. 6 were adequate for this analysis, therefore they were used with no modifications and are given in Table 3.8.

Fuselage Equations and Derivatives

Force and Moment (Trim)

$$X_F = -D_F \cos(\theta - \gamma_c) + L_F \sin(\theta - \gamma_c)$$

$$Y_F = S.F._F \cos \beta - D_F \sin \beta \cong 0$$

$$Z_F = -L_F \cos(\theta - \gamma_c) - D_F \sin(\theta - \gamma_c)$$

$$M_F = q \left(\frac{M}{q} \right)_F = q \frac{\partial (M/q)}{\partial \alpha_F} \alpha_F$$

$$N_F = q \left(\frac{N}{q} \right)_F$$

$$R_F = q \left(\frac{R}{q} \right)_F$$

Nondimensional Fuselage Derivatives

$$\frac{\partial \alpha_F}{\partial \dot{y}} = \frac{\partial \beta}{\partial \dot{y}} = -\frac{\partial \gamma}{\partial \dot{z}} = \frac{1}{V}$$

$$\left. \begin{array}{c} \dfrac{\partial f}{\partial \alpha_F} \\[1em] \dfrac{\partial L/q}{\partial \alpha_F} \\[1em] \dfrac{\partial Y/q}{\partial \beta} \\[1em] \dfrac{\partial M/q}{\partial \alpha_F} \end{array} \right\}$$
From curves in Appendix A of Ref. 6 at trim conditions

Continued

Table 3.8

$$\left.\begin{array}{c} \dfrac{\partial N/q}{\partial \beta} \\[2em] \dfrac{\partial R/q}{\partial \beta} \end{array}\right\} \quad \text{From curves in Appendix A of Ref. 6 at trim conditions}$$

Fuselage Derivatives

$$\left(\frac{\partial X}{\partial \dot x}\right)_F = \frac{2\,X_F}{V}$$

$$\left(\frac{\partial X}{\partial \dot z}\right)_F = \left(L_F - q\frac{\partial f}{\partial \alpha_F} \right) \frac{\partial \alpha_F}{\partial \dot z}$$

$$\left(\frac{\partial Y}{\partial \dot y}\right)_F = \frac{1}{V}\!\left(q\frac{\partial (Y/q)}{\partial \beta} - D_F\right)$$

$$\left(\frac{\partial Z}{\partial \dot x}\right)_F = \frac{2\,Z_F}{V}$$

$$\left(\frac{\partial Z}{\partial \dot z}\right)_F = \left(-D_F - q\,\frac{\partial L/q}{\partial \alpha_F}\right)\frac{\partial \alpha_F}{\partial \dot z}$$

$$\left(\frac{\partial R}{\partial \dot y}\right)_F = q\frac{\partial (R/q)}{\partial \beta}\frac{\partial \beta}{\partial \dot y}$$

$$\left(\frac{\partial M}{\partial \dot x}\right)_F = \frac{2}{V}M_F + q\,\frac{\partial (M/q)}{\partial \alpha_F}\frac{\partial \alpha_F}{\partial \dot x}$$

$$\left(\frac{\partial M}{\partial \dot z}\right)_F = q\,\frac{\partial (M/q)}{\partial \alpha_F}\frac{\partial \alpha_F}{\partial \dot z}$$

$$\left(\frac{\partial N}{\partial \dot y}\right)_F = q\,\frac{\partial (N/q)}{\partial \beta}\frac{\partial \beta}{\partial \dot y}$$

Table 3.8 (continued)

2.   Wing Derivatives

The dominant aerodynamic component to be included in the derivatives (for a tiltrotor in airplane mode) is the wing. The wing is the primary Z-force contributor and is a major X-force contributor in cruise. There is a need for a more detailed analysis with the helicopter

29

relations given in Ref. 6. Ref. 6 did not give any wing derivatives, however, Ref. 7 took the relations given in Ref. 6 for a Horizontal tail section and modified them for a wing. This works fine since they are basically both simple airfoils. The JANRAD helicopter derivatives were used with the main rotor influence terms removed. The only relations that were added for a tiltrotor were the control derivatives. These relationships were taken from Ref. 5 and/or derived using standard aerodynamic relationships. The resulting wing derivatives are depicted in Table 3.9.

<p align="center"><u>Wing Equations and Derivatives</u></p>

<u>Force and Moment (Trim)</u>

$$L_w = qA_w a_w (\theta + i_w - \varepsilon_{FW} - \gamma_c - \alpha_{ol}) \qquad \Rightarrow \quad C_{Lw} = L_w / (qA_w)$$

$$D_w = qA_w \left[\frac{C_{Lw}^2 (1 + \delta_{iw})}{\pi AR} + C_{Dow} \right]$$

$$X_w = -D_w \cos(\theta - \varepsilon_{FW} - \gamma_c) + L_w \sin(\theta - \varepsilon_{FW} - \gamma_c)$$

$$Z_w = -L_w \cos(\theta - \varepsilon_{FW} - \gamma_c) - D_w \sin(\theta - \varepsilon_{FW} - \gamma_c)$$

<u>Nondimensional Derivatives</u>

$$\frac{\partial \varepsilon_{Fw}}{\partial \dot{z}} = -\left(\frac{\partial \varepsilon_{Fw}}{\partial \alpha_F} \right)_w \frac{\partial \gamma}{\partial \dot{z}}$$

$$\frac{\partial \alpha_w}{\partial \dot{z}} = -\left(\frac{\partial \varepsilon_{Fw}}{\partial \dot{z}} + \frac{\partial \gamma}{\partial \dot{z}} \right)$$

<u>Wing Derivatives</u>

$$\left(\frac{\partial X}{\partial \dot{x}} \right)_w = \frac{2 X_w}{V}$$

$$\left(\frac{\partial X}{\partial \dot{z}} \right)_w = qA_w a_w \left\{ \left[1 - \frac{2a_w (1 + \delta_{iw})}{\pi AR} \right] (\alpha_w - \alpha_{ol}) + (\alpha_w - \iota_w) \right\} \frac{\partial \alpha_w}{\partial \dot{z}}$$

$$\left(\frac{\partial X}{\partial \delta_a} \right)_w = \frac{2qA_w^2 (1 + \delta_{iw})}{\pi b_w^2} \left(\frac{\partial C_{lw}}{\partial \delta_a} \right)^2 \delta_{flap}$$

Continued

<p align="center">Table 3.9</p>

<u>Wing Derivatives (continued)</u>

$$\left(\frac{\partial Z}{\partial \dot{x}}\right)_{\text{w}} = \frac{2\,Z_{\text{w}}}{V}$$

$$\left(\frac{\partial Z}{\partial \dot{z}}\right)_{\text{w}} = -qA_{\text{w}}a_{\text{w}}\left\{\frac{a_{\text{w}}(1+\delta_{\text{iw}})}{\pi AR}[2(\alpha_{\text{w}} - \alpha_{\text{olw}})(\alpha_{\text{w}} - \iota_{\text{w}}) + (\alpha_{\text{w}}\alpha_{\text{ol}})^2] + C_{\text{Dow}} + 1\right\}\frac{\partial\alpha_{\text{w}}}{\partial\dot{z}}$$

$$\left(\frac{\partial Z}{\partial q}\right)_{\text{w}} = -\left(\frac{\partial Z}{\partial \dot{z}}\right)_{\text{w}} l_{\text{w}}$$

$$\left(\frac{\partial R}{\partial p}\right)_{\text{w}} = \left(\frac{\partial Z}{\partial \dot{z}}\right)_{\text{w}}\left(\frac{b_{\text{w}}}{3}\right)^2$$

$$\left(\frac{\partial R}{\partial r}\right)_{\text{w}} = -\left(\frac{\partial Z}{\partial \dot{x}}\right)_{\text{w}}\left(\frac{b_{\text{w}}}{3}\right)^2$$

$$\left(\frac{\partial R}{\partial \delta_{\text{a}}}\right)_{\text{w}} = qA_{\text{w}}b_{\text{w}}\frac{\partial C_l}{\partial \delta_{\text{a}}}$$

$$\left(\frac{\partial M}{\partial \dot{x}}\right)_{\text{w}} = -\left(\frac{\partial X}{\partial \dot{x}}\right)_{\text{w}} h_{\text{w}} + \left(\frac{\partial Z}{\partial \dot{x}}\right)_{\text{w}} l_{\text{w}}$$

$$\left(\frac{\partial M}{\partial \dot{z}}\right)_{\text{w}} = -\left(\frac{\partial X}{\partial \dot{z}}\right)_{\text{w}} h_{\text{w}} + \left(\frac{\partial Z}{\partial \dot{z}}\right)_{\text{w}} l_{\text{w}}$$

$$\left(\frac{\partial M}{\partial q}\right)_{\text{w}} = \left(\frac{\partial Z}{\partial q}\right)_{\text{w}} l_{\text{w}}$$

$$\left(\frac{\partial N}{\partial p}\right)_{\text{w}} = \left(\frac{\partial X}{\partial \dot{z}}\right)_{\text{w}}\left(\frac{b_{\text{w}}}{3}\right)^2$$

$$\left(\frac{\partial N}{\partial r}\right)_{\text{w}} = \left(\frac{\partial X}{\partial \dot{x}}\right)_{\text{w}}\left(\frac{b_{\text{w}}}{3}\right)^2$$

$$\left(\frac{\partial N}{\partial \delta_{\text{a}}}\right)_{\text{w}} = \frac{qA_{\text{w}}(1+\delta_{\text{iw}})}{2\pi AR}\frac{\partial C_{l\text{w}}}{\partial \delta_{\text{a}}}$$

Table 3.9 (continued)

3. Horizontal Tail

The horizontal tail, (H-stab) for analysis purposes, is nothing more than another wing further aft from the center of gravity. For this reason, there are only a few subtle differences in the derivatives from the wing derivatives. First, JANRAD allowed for the difference in dynamic

31

pressure between the wing and the tail with the paramer, q_H/q. This parameter will remain in the titlrotor analysis. Second, most airplane designs have symetrical H-stabs which have zero lift at zero angle of attack. Allowing $\alpha_{ol(H)} = 0$, here, simplifies the wing equations somewhat, resulting in the ones shown in Table 3.10. The final differences from the wing derivatives apply the fact that the H-stab span, b_H is much shorter than the wing's, so the lateral derivatives for the R and N moments are small. Compared to the wing derivatives, these are essentially zero and are therefore excluded.

Horizontal Tail Equations and Derivatives

<u>Force and Moment (Trim)</u>

$$L_H = \left(\frac{q_H}{q}\right) q A_H a_H \overbrace{(\theta + i_H - \varepsilon_{FH} - \gamma_c)}^{\alpha_H} \qquad \Rightarrow \quad C_{L_H} = \frac{L_H}{q_H A_H}$$

$$D_H = \left(\frac{q_H}{q}\right) q A_H \left[\frac{C_{L_H}^2 \left(1 + \delta_{i_H}\right)}{\pi AR} + C_{Do_H} \right]$$

$$X_H = -D_H \cos(\theta - \varepsilon_{FH} - \gamma_c) + L_H \sin(\theta - \varepsilon_{FH} - \gamma_c)$$

$$Z_H = -L_H \cos(\theta - \varepsilon_{FH} - \gamma_c) - D_H \sin(\theta - \varepsilon_{FH} - \gamma_c)$$

<u>Nondimensional Derivatives</u>

$$\frac{\partial \varepsilon_{FH}}{\partial \dot{z}} = -\left(\frac{\partial \varepsilon_{FH}}{\partial \alpha_F}\right)_H \frac{\partial \gamma}{\partial \dot{z}}$$

$$\frac{\partial \alpha_H}{\partial \dot{z}} = -\left(\frac{\partial \varepsilon_{FH}}{\partial \dot{z}} + \frac{\partial \gamma}{\partial \dot{z}}\right)$$

<u>Dimensional Derivatives</u>

$$\left(\frac{\partial X}{\partial \dot{x}}\right)_H = \frac{2 X_H}{V}$$

$$\left(\frac{\partial X}{\partial \dot{z}}\right)_H = \left(\frac{q_H}{q}\right) q A_w a_w \left\{\left[1 - \frac{2 a_H (1 + \delta_{i_H})}{\pi AR}\right] \alpha_H + (\alpha_H - \iota_H)\right\} \frac{\partial \alpha_H}{\partial \dot{z}}$$

$$\left(\frac{\partial Z}{\partial \dot{x}}\right)_H = \frac{2 Z_H}{V}$$

Continued

Table 3.10

$$\left(\frac{\partial Z}{\partial \dot{z}}\right)_H = -\left(\frac{q_H}{q}\right) q A_H a_H \left\{ \frac{a_H (1 + \delta_{iw})}{\pi AR} [2\alpha_H (\alpha_H - \iota_H)] + C_{DoH} + 1 \right\} \frac{\partial \alpha_H}{\partial \dot{z}}$$

$$\left(\frac{\partial Z}{\partial q}\right)_H = -\left(\frac{\partial Z}{\partial \dot{z}}\right)_H l_H$$

$$\left(\frac{\partial Z}{\partial \delta_e}\right)_H = -\left(\frac{q_H}{q}\right) q A_w \frac{\partial C_{l_H}}{\partial \delta_e}$$

$$\left(\frac{\partial M}{\partial \dot{x}}\right)_H = -\left(\frac{\partial X}{\partial \dot{x}}\right)_H h_H + \left(\frac{\partial Z}{\partial \dot{x}}\right)_H l_H$$

$$\left(\frac{\partial M}{\partial \dot{z}}\right)_H = -\left(\frac{\partial X}{\partial \dot{z}}\right)_H h_H + \left(\frac{\partial Z}{\partial \dot{z}}\right)_H l_H$$

$$\left(\frac{\partial M}{\partial q}\right)_H = \left(\frac{\partial Z}{\partial q}\right)_H l_H$$

$$\left(\frac{\partial M}{\partial \delta_e}\right)_H = -\left(\frac{q_H}{q}\right) q A_w l_H \frac{\partial C_{l_H}}{\partial \delta_e}$$

Table 3.10 (continued)

4.   Vertical Tail

The final set of derivatives to be developed are also not that different from the previous sections. A vertical tail is also not much different from a wing positioned vertically. For an airplane, they usually have a full span rudder causing a change in the side force (side lift) in the same manner a flap or aileron change the lift on a wing. A twin vertical tail has the same effect as a single tail with twice the reference area, A_v, when interference and proprotor effects are neglected, therefore there is no need to complicate the analysis with trying to deal with multiple vertical tails. The additon of a control surface (rudder) is the only real difference in these relations, shown in Table 3.11 have from Ref. 6.

33

Vertical Tail Equations and Derivatives

Force (Trim)

$$L_v = \left(\frac{q_v}{q}\right) qA_v a_v(\beta) \approx 0 \qquad \Rightarrow \quad C_{Lv} \approx 0$$

$$D_v = \left(\frac{q_v}{q}\right) qA_v \left[\frac{C_{Lv}^2(1+\delta_{iv})}{\pi AR_v} + C_{Dov}\right] = \left(\frac{q_v}{q}\right) qA_v C_{Dov}$$

$$X_v = -D_v \cos(\theta - \varepsilon_{FH} - \gamma_c)$$

$$Y_v \approx 0$$

$$Z_v \approx 0$$

Nondimensional Derivatives

$$\frac{\partial\beta}{\partial\dot y} = -\frac{\partial\gamma}{\partial\dot z} = \frac{1}{V}$$

$$\frac{\partial M_F}{\partial\dot y} = \frac{\partial\eta_F}{\partial\beta}\frac{\partial\beta}{\partial\dot y}$$

$$\frac{\partial\alpha_v}{\partial\dot y} = -\left(\frac{\partial\beta}{\partial\dot y} + \frac{\partial\eta_F}{\partial\dot y}\right)$$

Vertical Tail Derivatives

$$\left(\frac{\partial Y}{\partial\dot y}\right)_v = \left(\frac{q_v}{q}\right) qA_v \frac{\partial\alpha_v}{\partial\dot y} a_v$$

$$\left(\frac{\partial Y}{\partial p}\right)_v = \left(\frac{\partial Y}{\partial\dot y}\right)_v h_v$$

$$\left(\frac{\partial Y}{\partial r}\right)_v = -\left(\frac{\partial Y}{\partial\dot y}\right)_v l_v$$

$$\left(\frac{\partial R}{\partial\dot y}\right)_v = \left(\frac{\partial Y}{\partial\dot y}\right)_v h_v$$

$$\left(\frac{\partial R}{\partial p}\right)_v = \left(\frac{\partial Y}{\partial p}\right)_v h_v$$

Continued

Table 3.11

<u>Vertical Tail Derivatives (continued)</u>

$$\left(\frac{\partial R}{\partial r}\right)_v = \left(\frac{\partial Y}{\partial r}\right)_v h_v$$

$$\left(\frac{\partial N}{\partial \dot{x}}\right)_v = -\left(\frac{\partial Y}{\partial \dot{x}}\right)_v \approx 0$$

$$\left(\frac{\partial N}{\partial \dot{y}}\right)_v = -\left(\frac{\partial Y}{\partial \dot{y}}\right)_v l_v$$

$$\left(\frac{\partial N}{\partial p}\right)_v = -\left(\frac{\partial Y}{\partial p}\right)_v l_v$$

$$\left(\frac{\partial N}{\partial r}\right)_v = -\left(\frac{\partial Y}{\partial r}\right)_v l_v$$

*** Control Inputs ***

$$\left(\frac{\partial Y}{\partial \delta_r}\right)_v = \left(\frac{q_v}{q}\right) q A_v \frac{\partial C_{l_v}}{\partial \delta_r}$$

$$\left(\frac{\partial N}{\partial \delta_r}\right)_v = -\left(\frac{\partial Y}{\partial \delta_r}\right)_v l_v$$

Table 3.11 (continued)

IV. MODEL VERIFICATION

A.   METHODOLOGY

Model verification is based on comparing JANRAD output for the stability derivatives and other data generated with data obtained from industry models of the XV-15 and the V-22. There are essentially four ways (feasible in this thesis) to verify or compare the JANRAD model with the tested models; these involve comparing the following stability analysis tools:

1. The stability and control derivatives themselves.
2. The roots of the plant (A) matrix.
3. Frequency response or Bode plots.
4. Time response to various inputs.

Because of the lack of availability of some detailed V-22 information, #2 is the only comparison feasible with the V-22. Comparisons 1, 3 and 4 are made with only the XV-15 model. The two tested models used for comparison come from two separate sources and they both have to be converted to the JANRAD format before they can be analyzed. The details about the source of the data and the data conversion process for each model are discussed in the following two sections.

1.   XV-15 Tiltrotor Model Data Conversion

The mathematical model for a Generic Tilt-Rotor System (GTRS) was developed by Bell Helicopter Textron (BHT) under a NASA contract. As discussed in Ref. 2, this model was made for real time simulation use on a VAX computer in support of aircraft design, pilot training and flight testing. The model was verified by applying XV-15 unique physical and aerodynamic characteristics and comparing the dynamics to XV-15 flight test data [Ref. 3] for trim conditions similar to those investigated by this thesis. Reference 2 is the mathematical model with the equations of motion and many aerodynamic relationships used in the GTRS model. This reference was used extensively in the development of the stability derivatives used in this research, along with detailing logical sign conventions that make the analysis of tiltrotor dynamics easier.

The primary source of information for our analysis is Ref. 4. Reference 4 is the result of the generic tiltrotor model run for two cases, an out of ground effect hover, and a 200 knot cruise

37

at a low altitude with the nacelles and rotor system in the airplane mode. The first set of information taken from Ref. 4 is the aircraft configuration and flight conditions. In order to perform an accurate comparison of the models, all these parameters should be as close as possible to each other. Most of these parameters were entered using JANRAD's input routine and saved as the Matlab data file, XV15H.MAT for the helicopter flight condition and XV15A.MAT for the cruise (airplane) conditions. All the conditions are depicted in Ref. 4 found in Appendix K.

All the trim parameters are also listed in Ref. 4 which for the helicopter mode proved to be very important. When the XV-15 hover configuration and flight conditions were applied to the JANRAD trim section, the trim parameters were not satisfactorily close to those in Ref. 4. Since the helicopter trim routine was not written by this project, its accuracy for a tiltrotor is questionable. Therefore, the hover model generated by this project was generated without the use of the JANRAD trim routine. The trim parameters were inserted (with a data file) to ensure both models had the same trim reference point. The airplane mode trimming routine written by this project did not have the same problem as the helicopter mode trim. Reference 4 airplane trim parameters matched fairly close to the results of the airplane trimming routine..

Once the trimming parameters were consistent, the resulting stability derivatives were compared. The model put through their verification process is based on a set of seven force and moment equations of motion, with the rotor rpm being the seventh state and rotor torque being the seventh (moment) equation. A helicopter is normally considered a constant rpm and (for a trim condition) constant torque machine. For small perturbation simulation modeling however, rotor rpm is a state that does have perturbations that effect the states in our state space and should not be considered constant. However, to remain consistent with JANRAD's structure, rotor rpm was not included in our state space. Our verification will be to compare our six degree of freedom model to theirs with the seventh (rotor rpm) removed. This is essentially the same as assuming the rotor rpm and torque to be constant, which is the assumption made in both Refs. 6 and 7.

Stability derivatives are generated in the GTRS model software using a small perturbation numerical routine described in Reference 4. The derivatives are not used in the simulation process but are calculated for analysis only. Derivatives are generated in both the body and rotor axes as depicted in Appendix K, but only body axis derivatives are used for comparison. To be compared directly, a separate routine was written to generate them because

38

JANRAD does not output the derivatives themselves. The GTRS generated stability and control derivatives for both modes were extracted from Ref. 4 and are compared with those found with JANRAD. The listing of all the stability and control derivatives is found in section B with Tables 4.1 and 4.2, respectively, for the hover mode and Tables 4.3 and 4.4 for airplane mode.

JANRAD displays the [A] and [B] matrices formed with derivatives and the appropriate moments of inertia, aircraft mass, etc. of the format shown in Appendix B [Ref. 8]. The GTRS generated stability derivatives shown as "Total" in the tables for both modes were extracted from Ref. 4 and also inserted into the same matrix of App. B to produce the A and B matrices for each flight regime. The matrices themselves could be compared directly but that would not be any more enlightening than comparing the stability and control derivatives. The matrices generated by JANRAD can be found in Appendices C and D, along with the JANRAD input and generated data.

2.   V-22 Osprey State Space Model Conversion

The V-22 data availability limits the degree of comparison that can be made for this project. The preferred method of comparison would be to attain the same physical characteristics that was used in developing the XV-15 model and use JANRAD to construct a model for the V-22. However, this information such as the aerodynamic relationships, component and center of gravity locations was not available.

The information that was available was an airframe state space model of the V-22. This model is one of many modules used to simulate the entire V-22 aircraft for the V-22 Manned Flight Simulator (MFS) at the Navy's Flight Test Center in Patuxent River, Maryland. The module applicable here is the airframe/rotor dynamics module, which contains not only the nine states in our model but also altitude and the longitudinal and lateral flapping angles for each rotor. The module contains all four matrices (A, B, C, and D of the classical form), however the A matrix is all that will be used in our comparison. Because this model is only one of many modules that represent all aspects of the V-22, the B matrix of the MFS model contains configurations that do not allow easy conversion to the JANRAD control input format of the four normal aircraft controls. Therefore, only the A matrix of this module is used, which corresponds to the state vector:

$$\mathbf{x}_{\text{af/rotor}} = [u, v, w, p, q, r, \Phi, \theta, \Psi, \text{alt}, a_{1s(L)}, a_{1s(R)}, b_{1s(L)}, a_{1s(R)}] \tag{4.1}$$

This state vector can be transformed to the JANRAD state vector format by eliminating the last five states, resulting in the state vector,

$$\mathbf{x'}_{af/rotor} = [u, v, w, p, q, r, \Phi, \theta, \Psi], \tag{4.2}$$

and then using a transformation matrix, $\mathbf{T}$, where $\mathbf{x}_{JANRAD} = \mathbf{T} \mathbf{x'}_{af/rotor}$. Eliminating the final five states is, again, essentially the same as assuming them to be constant, which is how JANRAD treats these parameters. Reducing the order of the A matrix by the same amount does not change the equations. The transformation matrix T was used to transform the A matrix into the JANRAD formatted matrix, A' as follows:

$$\dot{\mathbf{x}}_{af/rotor} = [A_{af/rotor}] \mathbf{x}_{af/rotor} + [B_{af/rotor}] \mathbf{u} \tag{4.3}$$

$$\dot{\mathbf{x}'}_{af/rotor} = [A'_{af/rotor}] \mathbf{x'}_{af/rotor} + [B_{af/rotor}] \mathbf{u} \tag{4.4}$$

$$\mathbf{T}^{-1} \dot{\mathbf{x}}_{JANRAD} = [A'_{af/rotor}] \mathbf{T}^{-1} \mathbf{x}_{JANRAD} + [B_{af/rotor}] \mathbf{u} \tag{4.5}$$

$$\dot{\mathbf{x}}_{JANRAD} = \mathbf{T} [A'_{af/rotor}] \mathbf{T}^{-1} \mathbf{x}_{JANRAD} + \mathbf{T} [B_{af/rotor}] \mathbf{u} \tag{4.6}$$

$$\dot{\mathbf{x}}_{JANRAD} = [A'] \mathbf{x}_{JANRAD} + \mathbf{T} [B_{af/rotor}] \mathbf{u} \tag{4.7}$$

This results in the transformation, $A' = \mathbf{T} [A'_{af/rotor}] \mathbf{T}^{-1}$. The transformed A' matrix is the source of comparison to the other models. The actual matrices that went through these transformations along with the resulting matrix are found in Appendix F. The control matrix is not included here because the control rigging relationships, as well as the control derivatives, were not available. The natural frequencies and the matrix itself are, therefore, the only real set of figures that can be used for comparison with the other models.

B.   DISCUSSION OF MODEL COMPARISON

1.   Stability Derivatives

XV-15 stability derivative information of Ref. 4 is conveniently divided into their two major contributors, rotor and airframe. The derivatives used in the GTRS model are the sum of

the two, giving the "Total" aircraft derivatives. The JANRAD hover mode model assumes the airframe effect to be negligible and, therefore, no airframe effect is included in the JANRAD model. Table 4.1 shows the breakdown of the GTRS derivatives in order to see how good JANRAD's assumption is.

Derivative	JANRAD	GTRS (XV-15) Model		
		Total	Rotor	Airframe
X_q	528.7	531.51	531.91	-0.4
X_u	-5.01	-5.13	-4.64	-0.49
X_w	-1.03	-1.09	-1.13	0.04
Y_p	-559.8	-506.58	-506.41	-0.17
Y_r	-9.38	-156.39	-159.08	2.69
Y_v	-5.01	-23.02	-5.92	-17.11
Z_q	0	108.14	129.6	-21.45
Z_u	0	-28.57	-29.4	0.83
Z_w	-77.16	-80.17	-77.42	-2.75
L_p	-23,430.04	-19,022.5	-24,515.78	5,493.28
L_r	58.09	6,152.93	6,632.97	-480.04
L_v	-49.72	-265.93	-237.54	-28.39
M_q	-5,134	-4,286.12	-4,232.63	-53.48
M_u	49.72	15.33	6.86	8.46
M_w	5.72	-0.46	6.81	-7.27
N_p	2,071	10,464.46	10,460.51	3.95
N_r	-1,294	-2,037.07	-1,974.26	-62.81
N_v	9.42	85.83	82.53	3.31

Table 4.1 Comparison of Hover Mode Stability Derivatives

As it can be seen from the GTRS hover model in Table 4.1, ten of the sixteen derivatives have negligible airframe effects where the lateral derivatives, L_p and Y_v, account for the most noticeable airframe effects. The Y_v effect can be explained by the fact that the airframe has more drag from a sideward wind than the rotor system has. All the airframe effects are basically due to the change in the airflow around the airframe due to small perturbations in the relative wind. This comparison leads to the conclusion that neglecting the airframe effects may not be a good assumption.

41

The more important comparison here is how the "Rotor" derivatives compare with those of JANRAD's. This analysis will determine how effective JANRAD's stability derivative equations are. With this analysis, only half of the hover model derivatives compare well, with the Z derivatives and L_r are the least comparable of the sixteen derivatives.

With the control derivatives shown in Table 4.2, neglecting the airframe effect in a hover proves to be a valid assumption for all but one derivative, $M_{\delta c}$. The pitching moment effect for the collective input is significant and clearly cannot be neglected. As far as comparing the control derivatives between models, they do not seem very comparable. With exception of the collective control to pitch derivative, they are all the correct sign and are of the same order of magnitude. Another observation is that the primary control derivatives (the derivatives of the intended response such as $X_{\delta e}$, $Y_{\delta a}$, $Z_{\delta c}$, $L_{\delta a}$, $M_{\delta e}$, and $N_{\delta p}$) are the more accurate and dominate in magnitude over the cross control derivatives. This may prove to make the overall model more accurate than it appears from these values.

Derivative	JANRAD	GTRS (XV-15) Model		
		Total	Rotor	Airframe
$X_{\delta e}$	253.1	537.46	537.46	0
$X_{\delta c}$	-28.75	-34.07	-34.07	0
$Y_{\delta a}$	-2.53	-17.53	-17.53	0
$Y_{\delta p}$	0	98.82	98.82	0
$Z_{\delta e}$	0	6.23	7.18	-0.95
$Z_{\delta c}$	-2,162	-2,164.63	-2,432.23	267.6
$L_{\delta a}$	13,850	12,753.15	13,526.8	-773.65
$L_{\delta p}$	0	1,099.79	1,093.81	5.98
$M_{\delta e}$	-2,513	-4,030.16	-4,030.76	0.61
$M_{\delta c}$	160.2	-60.89	104.62	-165.51
$N_{\delta a}$	-180.6	-1,694.22	-1,694.23	0.01
$N_{\delta p}$	3,102	6,643.45	6,643.45	0

Table 4.2 Comparison of Hover Mode Control Derivatives

The airplane model has problems similar to those of the helicopter model. Table 4.3 shows the comparison between JANRAD stability derivatives and the GTRS generated ones. Because the airplane mode stability derivatives are based solely on the airframe effects, the

GTRS airframe contribution is listed next to the total derivatives in the table. From this table, it is clear that assuming the rotor derivatives to be negligible in the airplane mode is not an accurate assumption. In eleven of the sixteen stability derivatives, the rotor contribution is clearly not negligible, and in 6 of these, the rotor contribution dominates the derivative. As far as a check of the JANRAD derivative equations, the table shows a good comparison between the JANRAD derivatives and the GTRS airframe derivatives.

Derivative	JANRAD	GTRS (XV-15) Model		
		Total	Airframe	Rotor
X_q	0	381.65	-9.13	390.78
X_u	-6.91	-167.22	-13.18	-154.04
X_w	23.48	29.44	28.04	1.4
Y_p	-257.7	-467.62	-384.02	-83.6
Y_r	1,629	3,684.5	1,474.79	2,209.72
Y_v	-105.4	-151.31	-119.34	-31.97
Z_q	-1,192	-4,983.29	-2,765.34	-2,217.95
Z_u	-77.13	-69.05	-64.89	-4.16
Z_w	-522.6	-487.88	-456.28	-31.6
L_p	-32,980	-41,002.57	-27,197.33	-13,805.24
L_r	11,090	-2,239.51	9,832.69	-12,072.2
L_v	-430.2	-676.91	-594.38	-82.53
M_q	-26,190	-44,624.14	-52,201.97	7,577.83
M_u	64.27	438.22	68.38	369.84
M_w	-794.9	-758.47	-913.43	154.96
N_p	4,223	-11,752.99	5,575.94	-17,328.92
N_r	-37,520	-67,257.17	-35,061.92	-32,195.25
N_v	1,164	655.53	870.31	-214.78

Table 4.3 Comparison of Airplane Mode Stability Derivatives

As expected, the control derivatives compared more favorably than the stability derivatives. Table 4.4 demonstrates that the rotor does play a negligible role in all the airframe control derivatives (δ_e , δ_a , and δ_p) and the expected dominate role in the power control (δ_c) derivatives. The one exception is the flaperon to yaw derivative, N_{δ_a} which shows an unexpected rotor influence to be dominant with the lateral input.

43

Derivative	JANRAD	GTRS (XV-15) Model		
		Total	Airframe	Rotor
$X_{\delta e}$	0	-26.52	-26.52	0
$X_{\delta c}$	2,020	2,064.32	43.67	2,020.65
$Y_{\delta a}$	0	1.67	0	1.67
$Y_{\delta p}$	-931.4	-1,095.47	-1,095.47	0
$Z_{\delta e}$	-1,159	-1,284.67	-1,284.67	0
$Z_{\delta c}$	0	24.87	-34.01	58.88
$L_{\delta a}$	18,620	16,946.88	15,614.62	1,332.25
$L_{\delta p}$	-3,360	-3,950.05	-3,950.05	0
$M_{\delta e}$	-25,470	-29,169.63	-29,169.63	0
$M_{\delta c}$	-3,948	-4,967.37	-94.48	-4,872.89
$N_{\delta a}$	-282.6	5,694.52	1,721.01	3,973.52
$N_{\delta p}$	21,240	25,679.72	25,679.72	0

Table 4.4 Comparison of Airplane Mode Control Derivatives

2. Plant Eigenvalues

The next model comparison to be made is analyzing the natural frequencies of both models. The uncoupled plants for the hover mode will be compared first, followed by the uncoupled plants of the airplane mode. All three model plants will be compared together for each of the two modes.

a. Hover Mode

The root analysis of both (longitudinal and lateral) models highlights that both JANRAD and GTRS models have similar root distributions. The V-22 roots, used as a sanity check, shows the same distribution. The longitudinal model root distribution is shown in Figure 4.1. It shows the oscillatory mode (similar to the airplane phugoid) being comparable in both frequency and damping. As for the non-oscillatory roots, the short period roots are not as close as the long period but they are both comparable.

JANRAD (XV-15) Longitudinal (Hover) Model Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.1360 + 0.3522i	-0.3602	0.3775
0.1360 - 0.3522i	-0.3602	0.3775
-0.5244	1.0000	0.5244
-0.1913	1.0000	0.1913

Compared with the GTRS (XV-15) Longitudinal (Hover) Model Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.0810 + 0.2352i	-0.3256	0.2487
0.0810 - 0.2352i	-0.3256	0.2487
-0.3733	1.0000	0.3733
-0.2005	1.0000	0.2005
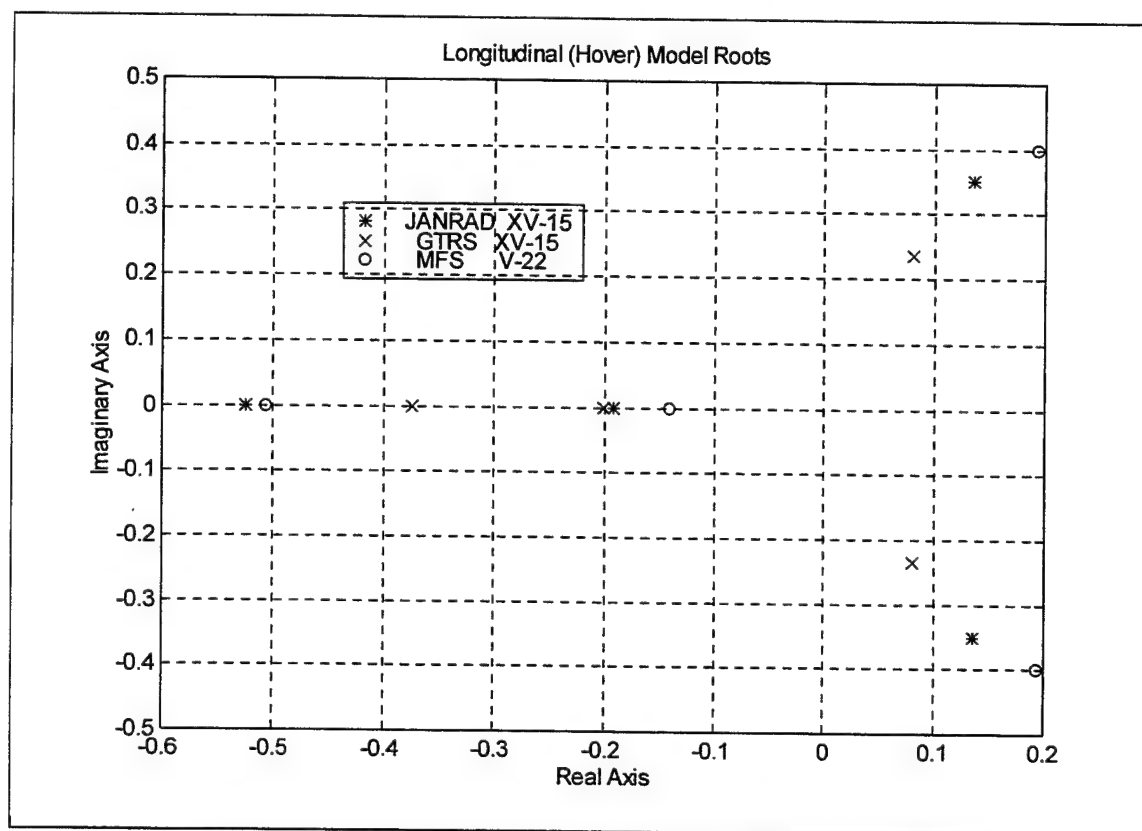
Figure 4.1 Comparison of Longitudinal (Hover) Models

And the MFS (V-22) Longitudinal (Hover) Model Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.1933 + 0.4004i	-0.4349	0.4446
0.1933 - 0.4004i	-0.4349	0.4446
-0.1416	1.0000	0.1416
-0.5065	1.0000	0.5065

45

The lateral roots depicted in Figure 4.2 are not as comparable as the longitudinal ones but do have comparable root distributions. The JANRAD oscillatory roots are within 50% of GTRS's frequency and damping. The real roots of the JANRAD model are closer to each other on the real axis than those of the GTRS model. These less comparable results are consistent with the stability derivative analysis of Section 1.

JANRAD Lateral (Hover) Model Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.0467 + 0.2302i	-0.1986	0.2349
0.0467 - 0.2302i	-0.1986	0.2349
-0.5485	1.0000	0.5485
-0.0199	1.0000	0.0199
0	-1.0000	0

Compared with the GTRS (XV-15) Lateral (Hover) Model Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.1445 + 0.4459i	-0.3083	0.4688
0.1445 - 0.4459i	-0.3083	0.4688
-0.7305	1.0000	0.7305
-0.0008	1.0000	0.0008
0	-1.0000	0
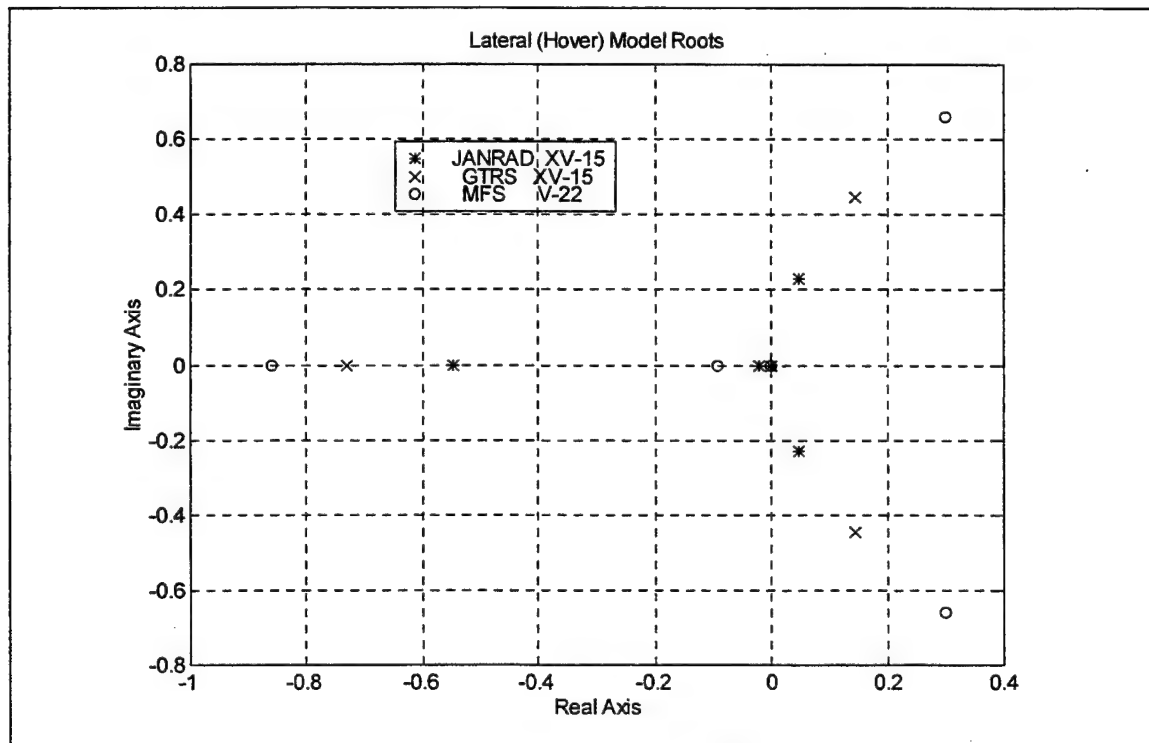
Figure 4.2 Comparison of Lateral (Hover) Models

And the V-22 Lateral (Hover) Model Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.3014 + 0.6583i	-0.4163	0.7241
0.3014 - 0.6583i	-0.4163	0.7241
-0.8587	1.0000	0.8587
-0.0918	1.0000	0.0918
0	-1.0000	0

b.   *Airplane Mode*

The models of the airplane mode have the roots shown in Figures 4.3 and 4.4. Again, the root analysis initially looks promising due to their distribution being so similar to each other. The longitudinal roots have similar shapes in that all three models each have two sets of oscillatory roots. The two XV-15 models compare very well with the frequencies of both (phugoid and short period) modes being within 5% of each other and the damping being very comparable as well.

JANRAD (XV-15) Longitudinal (Airplane) Model Roots:

Eigenvalue	Damping	Freq. (rad/sec)
-1.2941 + 3.5888i	0.3392	3.8150
-1.2941 - 3.5888i	0.3392	3.8150
-0.1951 + 0.1703i	0.7533	0.2590
-0.1951 - 0.1703i	0.7533	0.2590
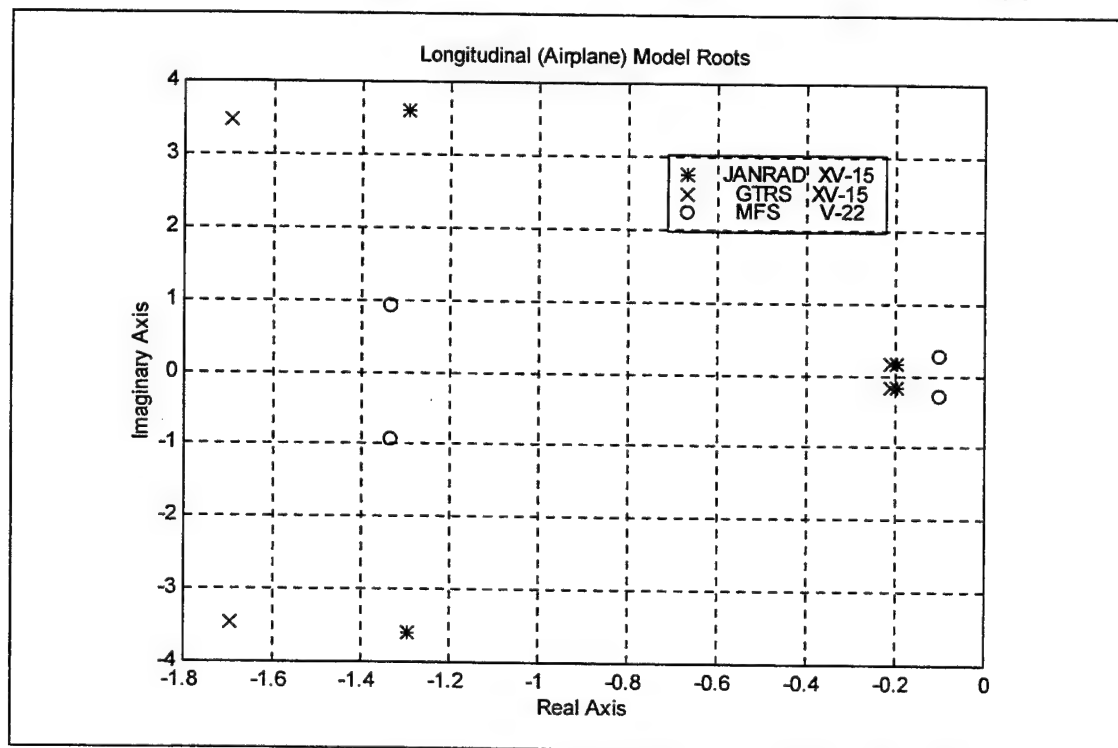
Figure 4.3 Comparison of Longitudinal (Airplane) Models

47

Compared with the GTRS (XV-15) Longitudinal (Airplane) Model Roots:

```
Eigenvalue          Damping      Freq. (rad/sec)
-1.6948 + 3.4555i   0.4403           3.8488
-1.6948 - 3.4555i   0.4403           3.8488
-0.2115 + 0.1576i   0.8018           0.2637
-0.2115 - 0.1576i   0.8018           0.2637
```

And the MFS (V-22) Longitudinal (Airplane) Model Roots:

```
Eigenvalue          Damping      Freq. (rad/sec)
-0.1002 + 0.2719i   0.3457           0.2898
-0.1002 - 0.2719i   0.3457           0.2898
-1.3340 + 0.9354i   0.8188           1.6293
-1.3340 - 0.9354i   0.8188           1.6293
```

The lateral roots again, do not fare as well as the longitudinal ones. As seen in the root listing, the models do compare with having the same number of oscillatory and pure real roots but the frequencies and damping could be closer. The JANRAD oscillatory roots (probably be Dutch roll) have a natural frequency 30% greater than that of the GTRS model and 60% the damping. The real roots do not compare favorably either with the JANRAD roots having significantly longer periods than the GTRS model.

JANRAD Lateral (Airplane) Model Roots:

```
Eigenvalue          Damping      Freq. (rad/sec)
-0.3958 + 2.3767i   0.1643           2.4094
-0.3958 - 2.3767i   0.1643           2.4094
-0.6645             1.0000           0.6645
-0.0058             1.0000           0.0058
      0            -1.0000              0
```

Compared with the GTRS (XV-15) Lateral (Airplane) Model Roots:

```
Eigenvalue          Damping      Freq. (rad/sec)
-0.4989 + 1.7702i   0.2712           1.8392
-0.4989 - 1.7702i   0.2712           1.8392
-1.0649             1.0000           1.0649
-0.1226             1.0000           0.1226
      0            -1.0000              0
```

And the V-22 Lateral (Airplane) Model Roots:

```
Eigenvalue          Damping      Freq. (rad/sec)
-0.1212 + 0.8156i   0.1470           0.8245
-0.1212 - 0.8156i   0.1470           0.8245
-1.0526             1.0000           1.0526
-0.1354             1.0000           0.1354
      0            -1.0000              0
```

48
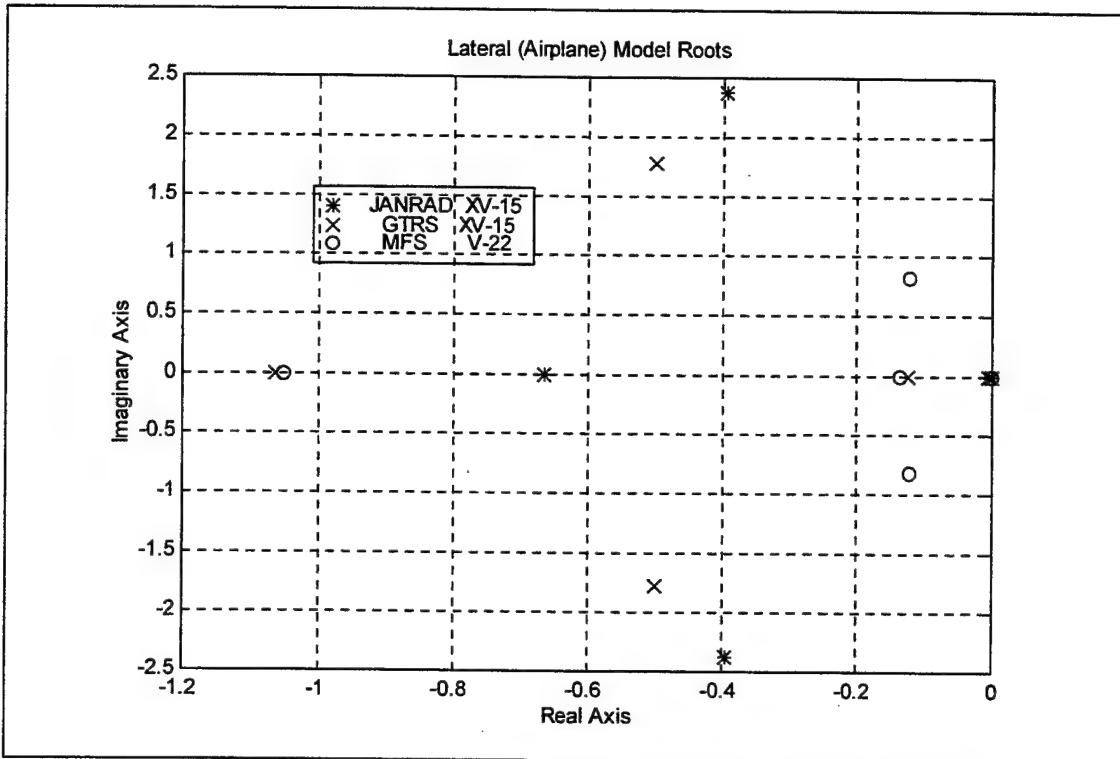
Figure 4.4 Comparison of Lateral Airplane Models (Roots)

3. Frequency Responses

A graphical comparison of the frequency domain analysis is the Bode plots. The old JANRAD produced Bode plots for all the applicable control inputs as one of its output routines. This routine was used to produce frequency responses for all states for each of the four inputs and are depicted listed in Appendix G for both hover and airplane models. The same plots were generated for the GTRS model and all the plots can be found in Appendix H. For the best comparison, the plots generated would preferably be done on the same set of axes or have the same scale, but with the number of plots generated, the given JANRAD plotting routines were used which produce the scale as shown. The appropriate input to desired output responses of the JANRAD model are compared to those of the GTRS model in the following subsections.

a. Hover Mode

The longitudinal plants compare very well as seen in Figures 4.5 and 4.6. Though the shapes of the two longitudinal cyclic responses are very close, the JANRAD model peaks at -12 dB where the GTRS model peaks at -4 dB. This 8 dB difference equates to a factor of about 6 meaning the GTRS model would respond six times more than the JANRAD model

49
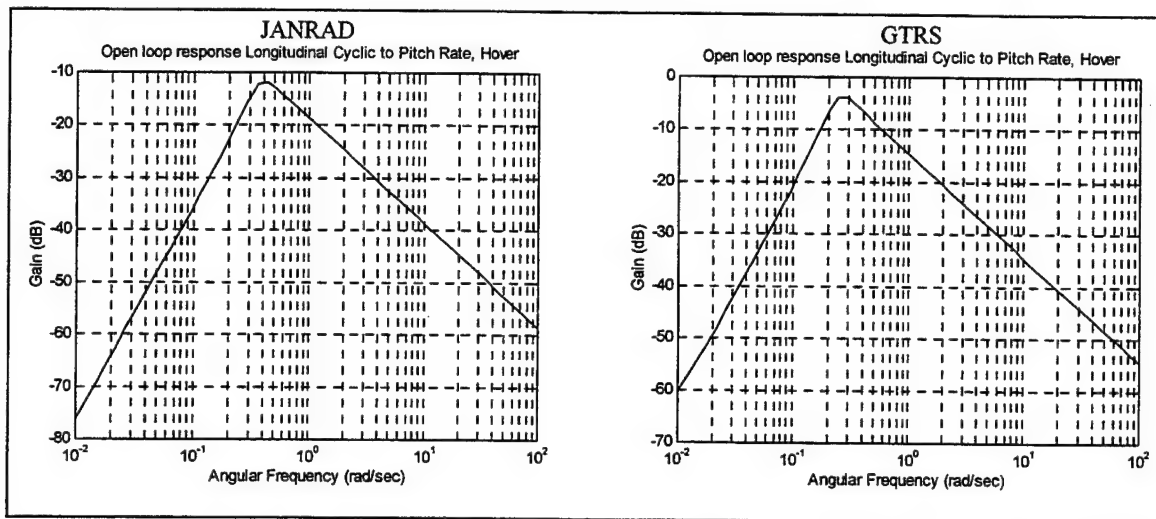
Figure 4.5 Comparison of Longitudinal Cyclic Frequency Responses, Hover

with an input of the same frequency (~0.3 rad/sec). The collective input responses are expectedly very close since the thrust derivatives and the masses of the two models are the same.
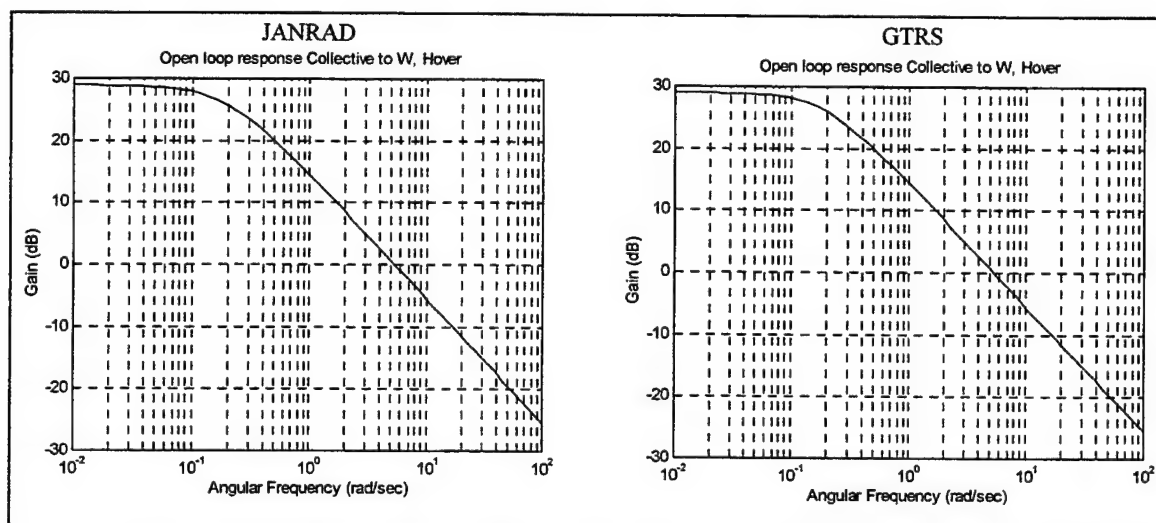
Figure 4.6 Comparison of Collective Frequency Responses, Hover

The lateral model responses of Figure 4.7 show slightly different shapes however, the major difference is in the trough seen in the low frequency range of the JANRAD response. This anomaly may be insignificant, however, due to how low the frequency is. The peak response is where the significant difference may show up. Here, the JANRAD response shows about a 8 dB higher peak response at a 2.5 rad/sec lower frequency. This indicates that the JANRAD hover model is more responsive to lateral inputs than the GTRS model.

50
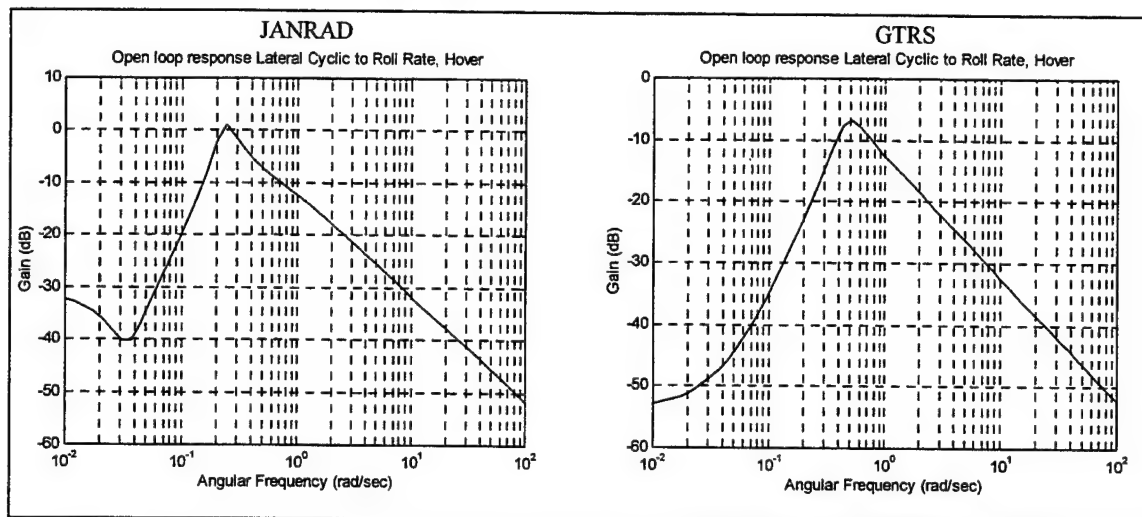
Figure 4.7 Comparison of Lateral Cyclic Frequency Responses, Hover

The directional pedal response comparison of Figure 4.8 shows nothing more than a frequency shift. This corresponds to a shift in magnitude as well since the frequency responses are linear (in dB) for most of the frequency range. The JANRAD model here, has a 7 dB lower response than the GTRS model.
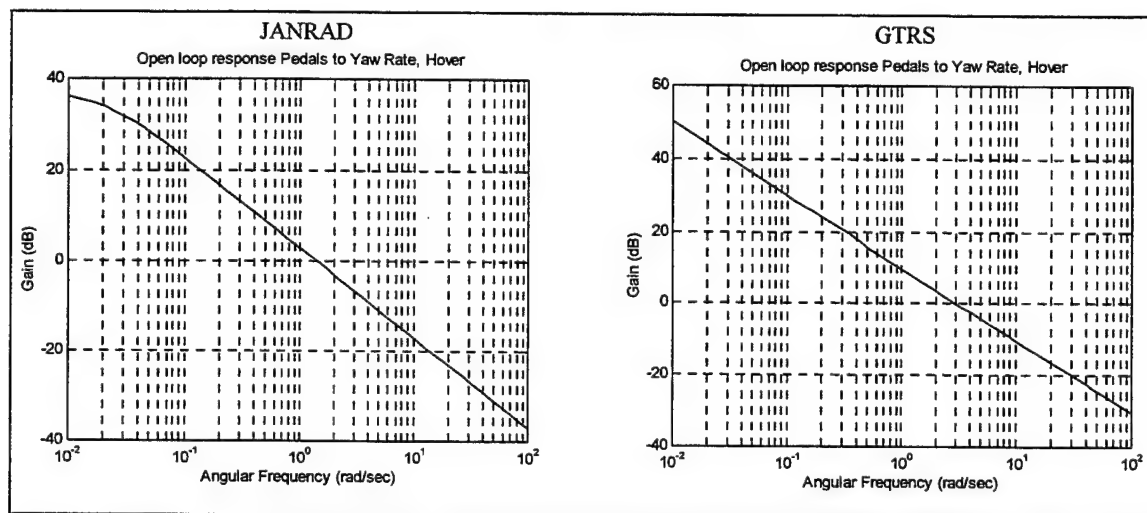
Figure 4.8 Comparison of Directional Pedals Frequency Responses, Hover

b. *Airplane Mode*

The airplane models performed somewhat better than the hover models. Figures 4.9 and 4.10 shows the comparison of the longitudinal cyclic and collective input responses respectively with very favorable results. In both comparisons, the frequency responses are almost indistinguishable.
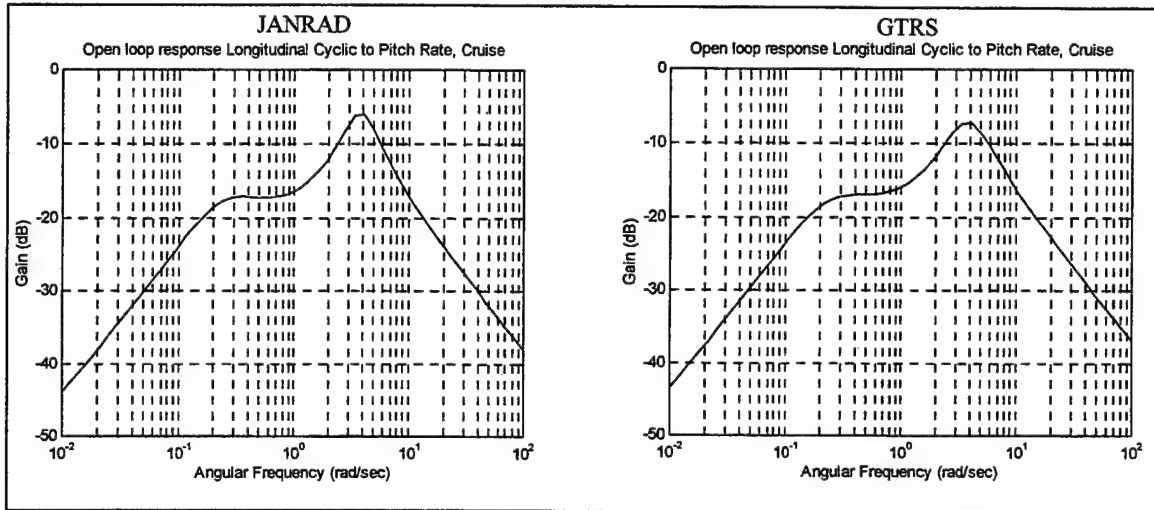
Figure 4.9 Comparison of Longitudinal Cyclic Frequency Responses, Airplane
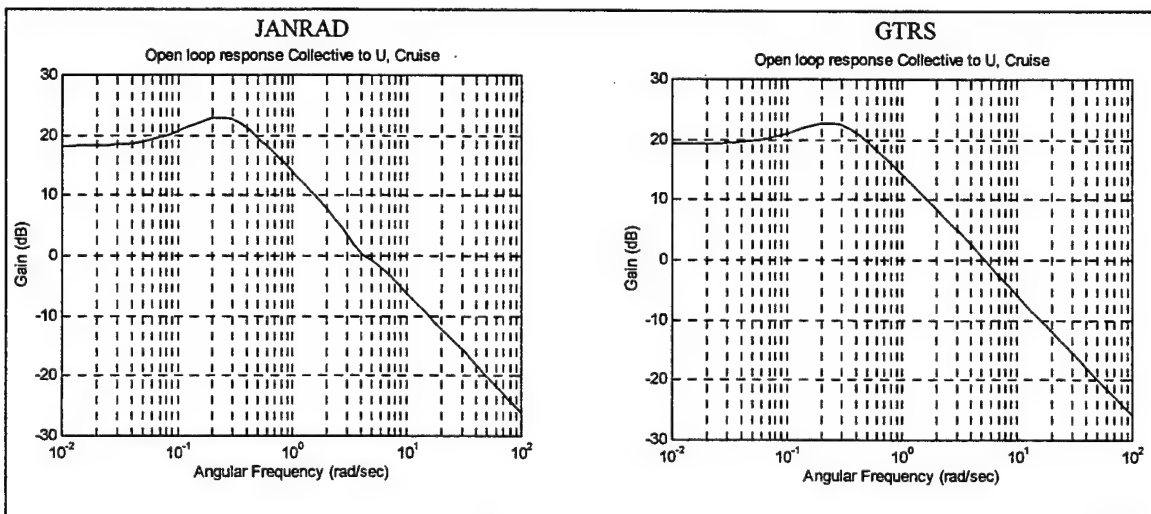
Figure 4.10 Comparison of Collective Frequency Responses, Airplane

The lateral frequency response comparisons shown in Figures 4.11 and 4.12 look favorable. The JANRAD lateral cyclic response shows a peak response of 3 dB more than the GTRS peak response at the lower break frequency but having virtually the same magnitude
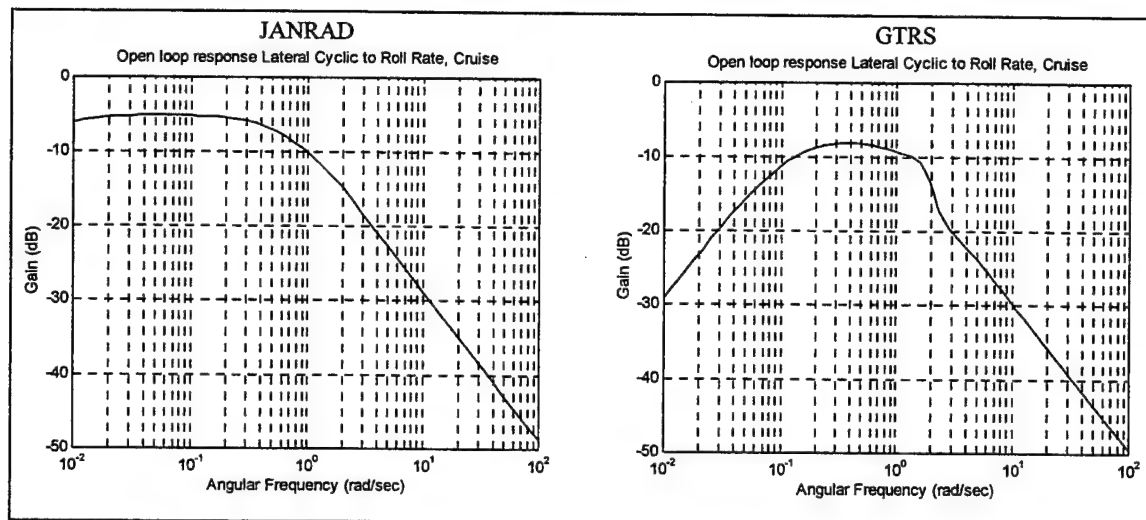
Figure 4.11 Comparison of Lateral Cyclic Frequency Responses, Airplane

for the frequencies above about 5 rad/sec. The pedal input responses have the same shape except for the lower frequencies. With close inspection of the peaks and troughs however, the comparison shows the magnitudes to be very close as well as the frequencies in which they occur.
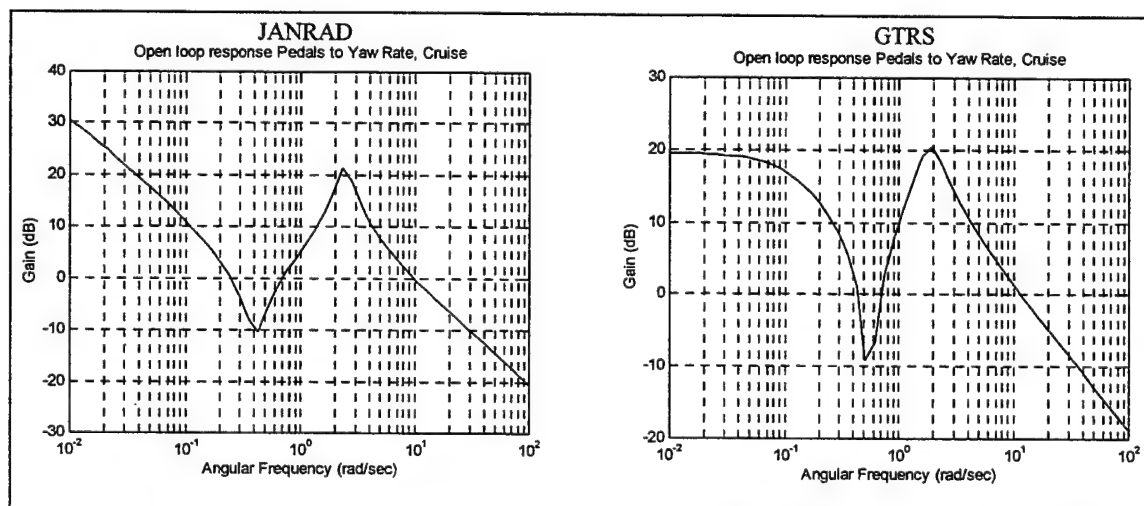
Figure 4.12 Comparison of Directional Pedals Frequency Responses, Airplane

4.   Time Responses

The models that JANRAD produce are merely trimmed condition ones which are not "true" dynamic models. Therefore, any type of time response using the MATLAB Control Systems Toolbox functions would not really depict any "true" dynamic response because very quickly after a disturbance, the trimmed condition has changed to where the model (plant) would no longer be valid. The time response output routine was added to JANRAD so that the initial response of a control input could be used for comparison and comparison only. As with the command bandwidth plots, time response plots were generated for all inputs and outputs for both the JANRAD models and the GTRS models and they can be found in Appendices I and J respectively. For all the inputs to their respective desired outputs, the time responses are shown and discussed in the following subsections.

a.   Hover Mode

The longitudinal models compare very well as seen in the time responses of Figures 4.13 and 4.14. The longitudinal cyclic response shows the JANRAD model to be less
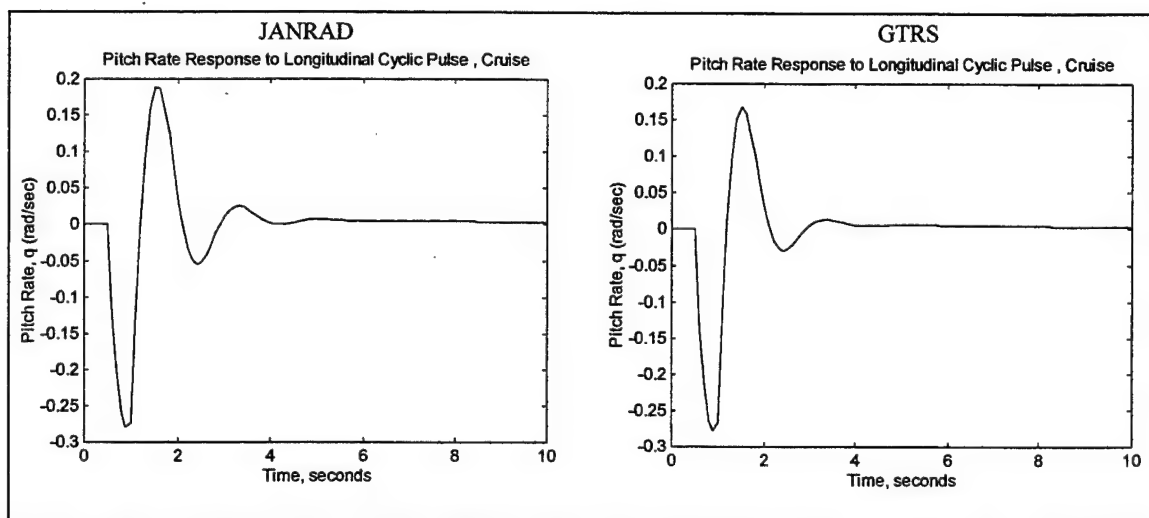
Figure 4.13 Comparison of Time Responses to Long. Cyclic Input , Hover

responsive with a -0.055 rad/sec (-3.15 deg/sec) max pitch rate when a 0.5 sec cyclic pulse was applied versus the -.09 rad/sec (5.16 deg/sec) pitch rate the GTRS model showed. The collective to vertical rate response in Figure 4.14 compared so well probably because the it is mostly due to the $\frac{\partial C_T/\sigma}{\partial \theta_o}$ derivative being the same between the models.   This thrust derivative (which essentially determines the change in thrust with a change in collective pitch) was taken directly
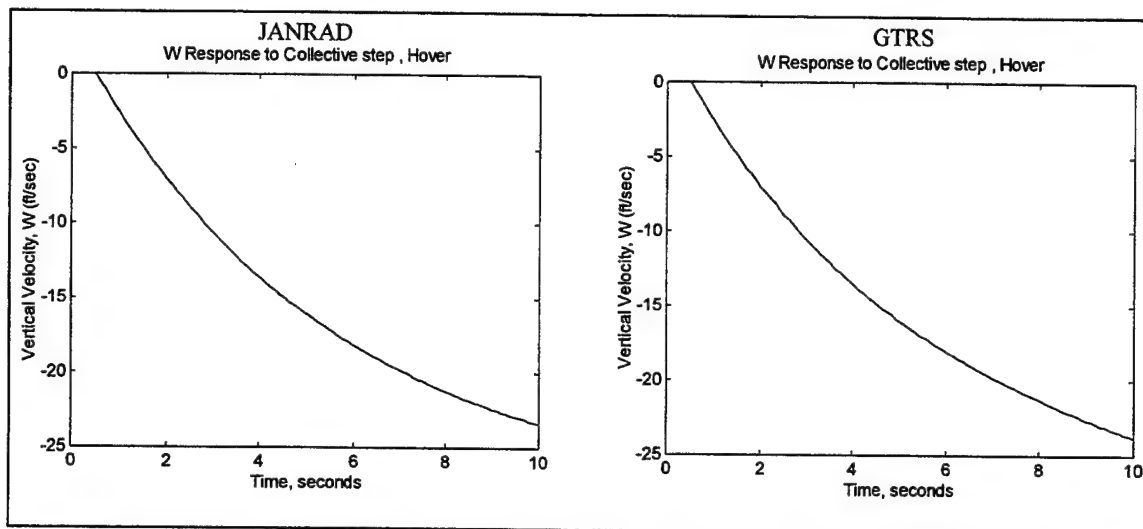
Figure 4.14 Comparison of Time Responses to Collective Input, Hover

from the XV-15 GTRS reference and "hardwired" in the code, therefore the time responses should be the same since the mass of the two aircraft models are the same.

The lateral models depicted in Figures 4.15 and 4.16 were not as comparable as the longitudinal ones. The max roll rate achieved for a lateral cyclic input were about the same at ~0.12 rad/sec (6.8 deg/sec), but the JANRAD model has a noticeable lower frequency
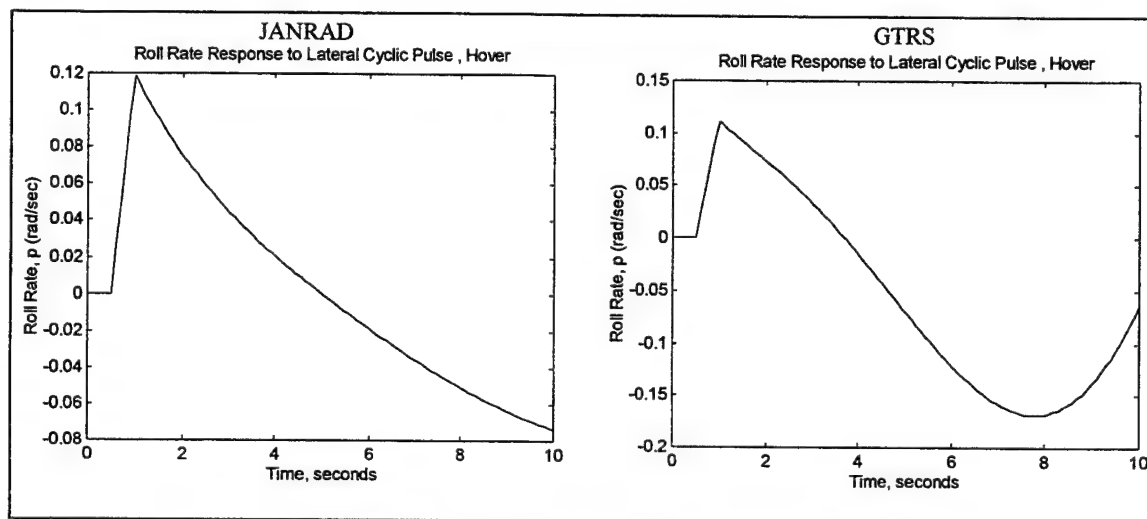
Figure 4.15 Comparison of Time Responses to Lateral Cyclic Input, Hover

response to the GTRS model. The pedal doublet response shows that the JANRAD model has about 50% the response to a pedal input the GTRS model has. These observations are consistent with the frequency response seen in Appendices G and H and discussed in the previous section.
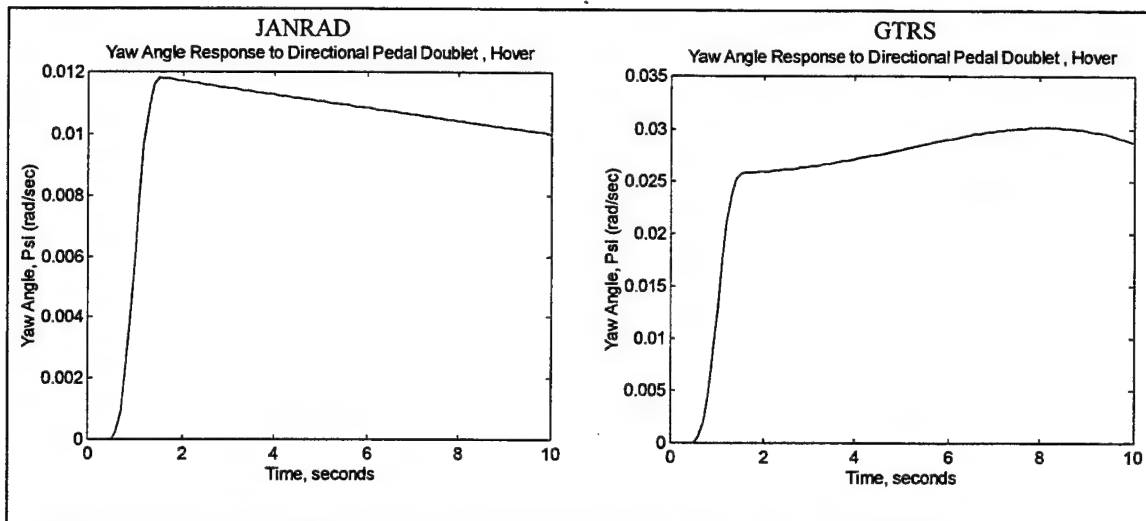
Figure 4.16 Comparison of Time Responses to Directional Pedal Input, Hover

a. *Airplane Mode*

The airplane mode longitudinal responses depicted in Figures 4.17 and 4.18 show very close matches between the models. This may conclude the concerns about the inconsistencies with the stability derivatives as well as the eigenvalues may not play as great a role as previously discussed. Both longitudinal and power input responses are almost indistiguishable between the JANRAD and GTRS plants.
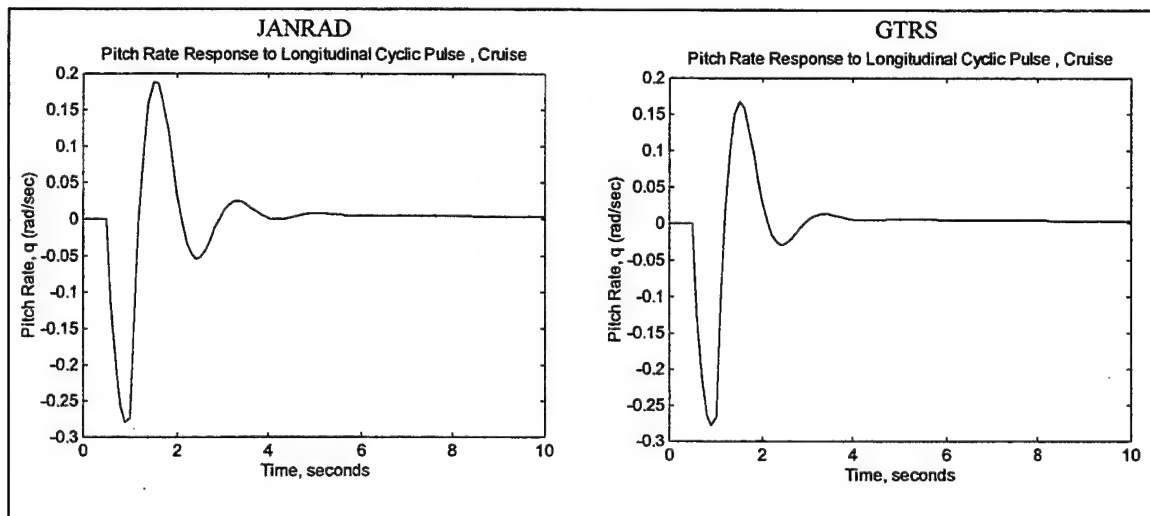
Figure 4.17 Comparison of Time Responses to Longitudinal Cyclic Input, Airplane

56
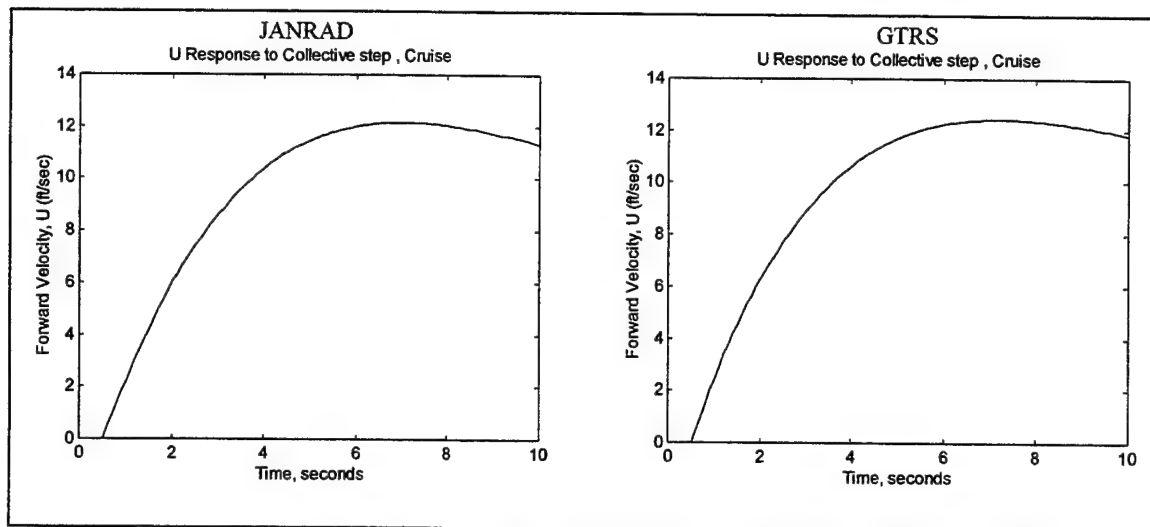
Figure 4.18 Comparison of Time Responses to Collective Input, Airplane

As seen with the frequency responses, the time response to a lateral stick input compared fairly well for the airplane model. Figure 4.19 shows the GTRS roll rate response
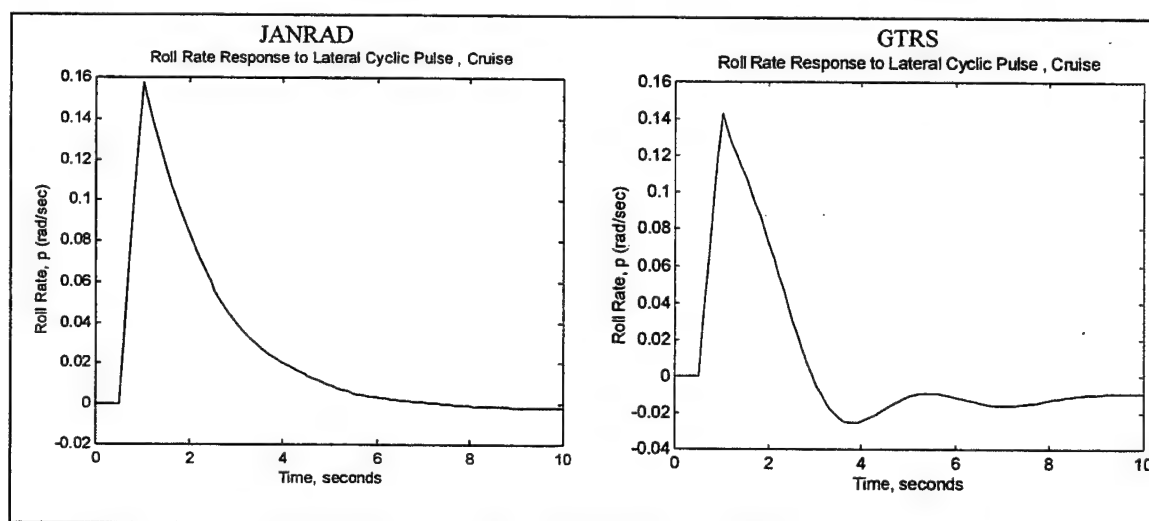
Figure 4.19 Comparison of Time Responses to Lateral Cyclic Input, Airplane

comparing very closely to that of the JANRAD's. The peak response is within 10 % and the only distinguishable difference is that JANRAD seems to be slightly overdamped where the GTRS model is slightly underdamped. Figure 4.20 shows the two models comparing very well in magnitude and damping when responding to the same pedal input.

57
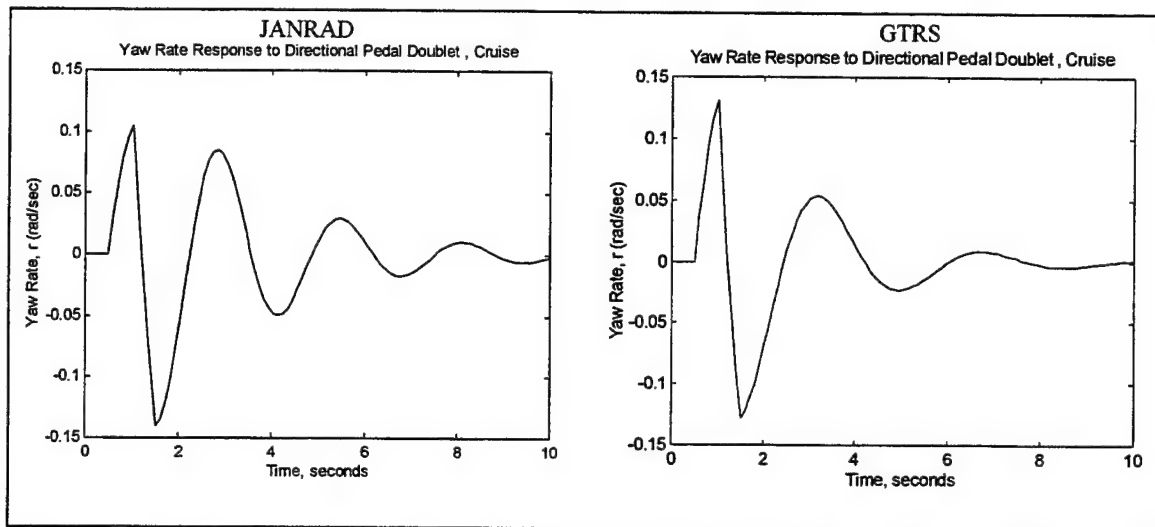
Figure 4.20 Comparison of Time Responses to Directional Pedal Input, Airplane

V. CONCLUSIONS AND RECOMMENDATIONS

A.   CONCLUSIONS

JANRAD has been verified to be a good preliminary design tool for helicopter performance analysis. The purpose of this thesis has been to evaluate the program for stability and control. JANRAD is an extensive program covering many areas of rotorcraft preliminary design to include performance, blade dynamics, and stability and control. The six degree of freedom state space model in JANRAD is unquestionably much simpler than the higher order models used to compare its results. However, its simplicity is one of its redeeming qualities. This project has attempted to verify its own modifications, and the results are fairly good. JANRAD was not originally written as a tiltrotor program. Therefore, it is not surprising that for this different type of aircraft, initial comparisons using helicopter-based stability and control derivatives did not look promising. The discrepancies involved the Prouty (Ref. 6) equations JANRAD uses and the assumptions about the fuselage effects in the hover mode and rotor effects in the airplane mode. These discrepancies point to the need for continued improvements in the modeling process.

The analysis of the root locations provided a clearer picture on the accuracy of the JANRAD generated model. This analysis demonstrated that both hover and airplane models were reasonably good when comparable modes in both uncoupled (longitudinal and lateral) plants were compared. Including the MFS model in the comparison proved valuable in verifying the plant/model conversion process to the JANRAD form. It also served as a good "sanity check" in that another tiltrotor (V-22) aircraft has the same basic modes as the JANRAD and GTRS XV-15 plants.

The other frequency domain analysis tool, Bode magnitude plots, provided the most information. In general, Bode plots compared well between the models. The longitudinal plants of both modes compared much better than the lateral plants. The frequency responses of the lateral modes show that the modeling equations should be improved. It was concluded that the lateral stability and control derivatives are the most likely source of the discrepancies encountered.

The final analysis tool used was a comparison of time histories. These comparisons showed promising results. Responses were similar if not indistiguishable in many of the comparisons. Any significant differences that occurred were in magnitude only. In some

59

comparisons there were slightly more or less damping, but in general, responses were comparable. Overall, the new tiltrotor routines) provide a fairly accurate assessment of the stability and control aspects of a tiltrotor in the preliminary stage of design.

B.   RECOMMENDATIONS

The first set of recommendations is to complete the goal of having a truly usable tiltrotor stability and control routine in JANRAD. This project accomplished most of this goal. The following are additional tasks needed to complete the work.

1.   Develop a tiltrotor performance section to:

a.   Incorporate and validate C_T/σ, etc. derivatives

b.   Develop a common trim routine for both tiltrotors and helicopters in all flight regimes.

2.   Add a program to input many "hardwired" data found in various routines

3.   Modify (long) stick movement to be consistent with helicopter section (aft=pos)

The JANRAD performance section has been verified with data obtained on an H-60 helicopter (Eccles, Ref. 9). Eccles showed excellent agreement but the helicopter methods did not work well with trimming a tiltrotor aircraft. The existing routine is used by the stability and control trim routine to provide trim swashplate positions for the main rotor and the tail rotor. This routine is also used iteratively to get derivatives such as C_T/σ needed in the stability and control section. This project used constant data found or calculated with data in Ref. 4. Existing performance routines) should be modified and then verified versus XV-15 or V-22 data to provide baseline performance data for tiltrotor aircraft.

The trim routine attempted to be used in this project, TILTTRIM.M, was one that was a modification of the previous routine, TRIM.M. Although this project did not use the TILTTRIM.M results, it does work for both helicopter and tiltrotor trimming. The routine was originally written for use in a tiltrotor design, but it was not written for general purpose. Modifications could be incorporated to make it fully functional in the JANRAD program for both types of rotorcraft.

Another set of modifications to make a truly functional stability and control section would be to incorporate a list of input parameters that would eliminate the "hardwired" data in the JANRAD routine. CBODYGRP.M, CTRGRP.M, and CTLTRGRP.M all have data such as

60

this in their first few lines of code that are not calculated or input. Most of these derivatives or constants are in these routines because they were taken from a plot, and an equation for them was not easily obtained. The problem is that they are only accurate for a particular type and size of aircraft and need to be modified for each design. These items could be incorporated using scaling factors that could be part of the input set of data or could even be calculated using the fuselage length or rotor radius depending upon the type of derivative in question.

The next set of recommendations are for improving on the accuracy of the tiltrotor modeling process. The stability and control derivative comparisons produced unfavorable results that could be improved. One improvement would be to include the airframe effects on the hover mode dynamics. This could be done by developing derivatives for the more influential airframe hover stability derivatives such as Y_v, L_p, M_u, and M_w and control derivatives such as $L_{\delta a}$ and $M_{\delta c}$. Another improvement would be to include the rotor influence in the airplane mode. This would be done by developing relationships for some of the more influential (in the airplane mode) rotor stability derivatives, such as X_q, X_u, Y_r, Z_q, L_p, L_r, M_u, N_p, and N_r, and the influential control derivatives such as $N_{\delta a}$. The incorporation of these derivatives would undoubtedly improve the accuracy of the JANRAD models.

One bit of design criterion that is used in both flying tiltrotor aircraft is the wing's dihedral and sweep. Both designs have positive dihedral for stability purposes and forward sweep that also affects their stability characteristics. These angles are easily accessible and also could be incorporated in the airframe stability derivatives if the proper relationship is used.

Another recommendation for further research related to the JANRAD program is to verify its results for an existing helicopter. Stability and control derivatives and entire plants for virtually all flight conditions are available for the H-60 as part of the NASA H-60 program. Similar to this project, parameters acquired from that program could be input using JANRAD, and the subsequent models/plants could be compared to the NASA results. This would be a method of validating the JANRAD stability and control section for a helicopter.

JANRAD is an effective tool for preliminary design, but it is not perfect and could always be improved. Its greatest attribute, however, is that it is easily updated and can be modified for future design needs.

APPENDIX A. JANRAD INPUT PARAMETERS LIST

JANRAD Input Parameters

Basic JANRAD
     PA = Pressure altitude (ft)
     temp = Temperature (deg F)
     Vinf = Airspeed (knots): ')*1.68894444;
     GW = Aircraft gross weight (lbs)
     b = Number of blades
     R = Blade radius: center of hub to blade tip (ft)
     rchord = Blade root chord (ft)
     e = Hinge offset (ft)
     grip = Non-aerodynamic inboard portion of blade (ft)
     twist = Blade twist (deg): ')/57.3;,twist=abs(twist);
     wblade = Weight of aero portion of one blade (lbs)
     nbe = Number of blade elements
     omega = Rotor rotational velocity (rad/sec)
     naz = Number of azimuth sectors
     a = Lift curve slope of rotor airfoil (CL vs alpha)
     Airfoil = Airfoil 1. HH-02   2. VR-12   3. NACA 0012
     thetao = Collective pitch at .7 r/R (deg): ')/57.3
     Afh = Aircraft equivalent flatplate area (ft^2)
     Afv = Vertical projected area (ft^2)
     Swing = Wing area, 0 if no wing (ft^2)
     bwing = Wing span (ft)
     CLwing = Expected CL for the wing
     CDowing = Wing profile drag coef (CDo)
     ewing = Wing efficiency factor (e)
     Shoriz = Horizontal tail area, 0 if none (ft^2)
     if Shoriz~=0,
       bhoriz = Horizontal tail span (ft)
     CLhoriz = Expected CL for the horizontal tail
     CDohoriz = Horizontal tail profile drag coef (CDo)
     Svert = Vertical tail area, 0 if none (ft^2)
       if Svert~=0,
         bvert = Vertical tail span (ft)
         CLvert = Expected CL for the vertical tail
         CDovert = Vertical tail profile drag coef (CDo)
     Taux = Auxiliary thrust (lbs)
     tr = Blade chord taper ratio (tip/chord)
     trst = Blade taper start position (r/R)

Stability & Control (with Tiltrotor) Routine

Main rotor
     Ib = Blade flapping moment of inertia (slug ft^2)
     hmd = Hub height above reference datum/waterline (ft)
     lmd = Hub fuselage station (ft)
     ymd = Hub position right of buttline (ft)
     im = Mast incidence (negative forward, deg)/57.3;
     Kflpsprng = Hub flapping spring constant (ft-lbs/deg)*57.3;

Tail rotor

        htd = Tail rotor height above reference datum/waterline (ft)
        ltd = Tail rotor fuselage station (ft)
        ytd = Tail rotor position right of buttline (ft)
        bt = Number of tail rotor blades
        cot = Tail rotor blade chord (ft)
        Rt = Tail rotor blade radius (ft)
        at = Average lift curve slope of tail rotor
        ohmt = Rotational velocity of tail rotor (rad/sec)
        Ibt = Tail rotor blade flapping moment of inertia (slug ft^2)
        delta3 = Delta-3 angle (deg)/57.3;
        theta1t = Blade twist (deg)/57.3;

NOTAR

        htnd = Height above reference datum/waterline (ft)
        ltnd = Fuselage station (ft)
        ytnd = Position right of buttline (ft)
        dian = NOTAR boom diameter (ft)
        swirl = Swirl angle at boom (deg)/57.3;
        Ytmaxn = Maximum NOTAR thruster force (lbs)
        lttnd = Thruster fuselage station reference (ft)

Tilt Rotor

        lfd = Fuselage station of (fuselage) Center of Pressure (ft)
        alplof = Fuselage angle @ zero lift (degrees) /57.3;
        af = Lift curve slope of fuselage (1/rad)
        cmof = Fus. mom. coef. @ zero alpha (ref. to Aw & cw)
        cmalpf = Slope of fus. moment coef. wrt alpha curve (1/rad)
        delih = Horizontal Tail Span Efficiency, (e) - 1;
        epso = Downwash angle @ zero alpha (rad)
        dclhddeleh = Change in H-stab Cl wrt elevator angle (1/rad)
        acw = Wing Aerodynamic Center location (% cw)
        lambda = Wing sweep angle (deg)*pi/180;
        dih = Wing Dihedral angle (deg)*pi/180;
        cmow = Wing Moment Coeff @ zero lift
        dclwddelf = Change in roll mom. coeff. wrt flaperon defl.(1/rad)

Vertical tail

        hvd = Height above reference datum/waterline (ft)
        lvd = Fuselage station (ft)
        yvd = Position right of buttline (ft)
        alplov = Zero lift angle for vertical tail (deg)/57.3;
        clvertmax = Maximum Cl for vertical tail
        qvq = Dynamic pressure ratio for tail (pg 489 Prouty)
        av = Lift curve slope of vertical tail
        crv = Vert. tail root chord (ft)
        ctv = Vert. tail tip chord (ft)
        delih=1/(input('Vert. tail span efficiency factor (e))-1;
        cfcv = Rudder chord length (% cv)

Horizontal tail

        hhd = Height above reference datum/waterline (ft)
        lhd = Fuselage station (ft)

64

yhd = Position right of buttline
        alploh = Zero lift angle for horizontal tail (deg)/57.3;
        ih = Angle of incidence of horizontal tail (deg)/57.3;
        ah = Lift curve slope of horizontal tail
        qhq = Dynamic pressure ratio for tail (pg 489 Prouty)
        vhvl = Rotor downwash ratio for h-tail (pg 489 Prouty)
        depsdalph = Fuselage downwash ratio for h-tail (pg 489 Prouty)

Wing
        hwd = Height above reference datum/waterline (ft)
        lwd = Fuselage station (ft)
        ywd = Position right of buttline (ft)
        alplow = Zero lift angle for wing (deg)/57.3;
        iw = Angle of incidence of wing (deg)/57.3;
        aw = Lift curve slope of wing
        ctw = Tip cord (ft)
        crw = Root cord (ft)
        vwvl = Rotor downwash ratio for wing (pg 489 Prouty)
        detafdalpfw = Fuselage downwash ratio for wing (pg 489 Prouty)

CG location
        zcg = CG height above reference datum/waterline (ft)
        xcg = CG Fuselage station (ft)
        ycg = CG position right of buttline (ft)

Fuselage moments of inertia/downwash parameter
        Ixx = Ixx (slug ft^2)
        Iyy = Iyy (slug ft^2)
        Izz = Izz (slug ft^2)
        Ixz = Ixz (slug ft^2)
        vfvl = Downwash ratio for fuselage (page 513 Prouty)

Rigging
        dblmddele = Long cyclic pitch per inch defl (deg/in)/57.3;
        dalmddela = Lateral cyclic pitch per inch defl (deg/in)/57.3
        dthetomddelc = Collective pitch per inch defl (deg/in)/57.3

    if notar==0
        dthetotddelp = Tail rotor pitch change per inch defl
                    or % twist   (deg/in or deg/deg of twist)/57.3
    if notar==1
      sidearm = Max deflection of anti-torque from neutral for NOTAR,
              enter 1000 if using tail rotor (deg or inch travel)*2
    if ctail==1
        maxr = Displacement of anti-torque control until full rudder
              deflection (deg or inch travel)
      dclvddelp = Change in (side force) lift wrt rudder defl. (1/rad)

    if tltrotr==1
      ddeladlat = Aileron angle per inch defl (deg/in)/57.3;
      dthetomddelc = Collective thrust per inch defl (deg/in)/57.3;
      ddeledlong = Elevator angle per inch defl (deg/in)/57.3;
      ddelrddelp = Rudder angle per inch defl of pedals/57.3;

65

For a the state vector, $\mathbf{x} = [\Delta u \ \Delta w \ \Delta q \ \Delta\theta \ \Delta v \ \Delta p \ \Delta\phi \ \Delta r \ \Delta\Psi]^T$

$$
A =
\begin{bmatrix}
\frac{X_u}{m} & \frac{X_w}{m} & \frac{X_q}{m}-w_o & -g\cos\theta_o & \frac{X_v}{m} & \frac{X_p}{m} & 0 & \frac{X_r}{m}+v_o & 0 \\[6pt]
\frac{Z_u}{m} & \frac{Z_w}{m} & \frac{Z_q}{m}+u_o & -g\cos\phi_o\sin\theta_o & \frac{Z_v}{m} & \frac{Z_p}{m}-v_o & -g\sin\phi_o\cos\theta_o & \frac{Z_r}{m} & 0 \\[6pt]
\frac{M_u}{I_y} & \frac{M_w}{I_y} & \frac{M_q}{I_y} & 0 & \frac{M_v}{I_y} & \frac{M_p}{I_y} & 0 & \frac{M_r}{I_y} & 0 \\[6pt]
0 & 0 & \cos\phi_o & 0 & 0 & 0 & 0 & -\sin\phi_o & 0 \\[6pt]
\frac{Y_u}{m} & \frac{Y_w}{m} & \frac{Y_q}{m} & -g\sin\phi_o\sin\theta_o & \frac{Y_v}{m} & \frac{Y_p}{m}+w_o & -g\cos\phi_o\cos\theta_o & \frac{Y_r}{m}-u_o & 0 \\[6pt]
\frac{(I_z L_u+I_{xz}N_u)}{I_c} & \frac{(I_z L_w+I_{xz}N_w)}{I_c} & \frac{(I_z Lq+I_{xz}N_q)}{I_c} & 0 & \frac{(I_z L_v+I_{xz}N_v)}{I_c} & \frac{(I_z L_p+I_{xz}N_p)}{I_c} & 0 & \frac{(I_z L_r+I_{xz}N_r)}{I_c} & 0 \\[6pt]
0 & 0 & \sin\phi_o\tan\theta_o & 0 & 0 & 1 & 0 & \cos\phi_o\tan\theta_o & 0 \\[6pt]
\frac{(I_{xz}L_u+I_x N_u)}{I_c} & \frac{(I_{xz}L_w+I_x N_w)}{I_c} & \frac{(I_{xz}Lq+I_x N_q)}{I_c} & 0 & \frac{(I_{xz}L_v+I_x N_v)}{I_c} & \frac{(I_{xz}L_p+I_x N_p)}{I_c} & 0 & \frac{(I_{xz}L_r+I_x N_r)}{I_c} & 0 \\[6pt]
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
$$

where, $I_c = I_x I_z - (I_{xz})^2$

For the control vector, $\mathbf{u} = [\Delta\delta_e \ \Delta\delta_c \ \Delta\delta_a \ \Delta\delta_p]^T$

$$
B =
\begin{bmatrix}
\frac{X_{\delta_e}}{m} & \frac{X_{\delta_c}}{m} & \frac{X_{\delta_a}}{m} & \frac{X_{\delta_p}}{m} \\[6pt]
\frac{Z_{\delta_e}}{m} & \frac{Z_{\delta_c}}{m} & \frac{Z_{\delta_a}}{m} & \frac{Z_{\delta_p}}{m} \\[6pt]
\frac{M_{\delta_e}}{I_y} & \frac{M_{\delta_c}}{I_y} & \frac{M_{\delta_a}}{I_y} & \frac{M_{\delta_p}}{I_y} \\[6pt]
0 & 0 & 0 & 0 \\[6pt]
\frac{Y_{\delta_e}}{m} & \frac{Y_{\delta_c}}{m} & \frac{Y_{\delta_a}}{m} & \frac{Y_{\delta_p}}{m} \\[6pt]
\frac{(I_z L_{\delta_e}+I_{xz}N_{\delta_e})}{I_c} & \frac{(I_z L_{\delta_c}+I_{xz}N_{\delta_c})}{I_c} & \frac{(I_z L_{\delta_a}+I_{xz}N_{\delta_a})}{I_c} & \frac{(I_z L_{\delta_p}+I_{xz}N_{\delta_p})}{I_c} \\[6pt]
0 & 0 & 0 & 0 \\[6pt]
\frac{(I_{xz}L_{\delta_e}+I_x N_{\delta_e})}{I_c} & \frac{(I_{xz}L_{\delta_c}+I_x N_{\delta_c})}{I_c} & \frac{(I_{xz}L_{\delta_a}+I_x N_{\delta_a})}{I_c} & \frac{(I_{xz}L_{\delta_p}+I_x N_{\delta_p})}{I_c} \\[6pt]
0 & 0 & 0 & 0
\end{bmatrix}
$$

APPENDIX C. JANRAD OUTPUT FOR XV-15 DESIGN HOVER MODE

```
*** RESULTS ***
      xv15h

*** INPUT DATA ***
```

Flight Conditions

```
        Forward velocity =        0 kts
             Temperature =       59 degs F
       Pressure altitude =        1 ft
        Auxiliary thrust =        0 lbs
```

Fuselage

```
                Gross weight =  13000 lbs
    Equivalent flat plate area =     2.5 ft^2
      Vertical projected area =   530.0 ft^2
   CG height above waterline =       6.8 ft
          CG fuselage station =      25.0 ft
   CG position rt of buttline =       0.0 ft
                         Ixx = 52795.0 slug ft^2
                         Iyy = 21360.0 slug ft^2
                         Izz = 66335.0 slug ft^2
                         Ixz =  1234.0 slug ft^2
             Downwash ratio =       1.00
```

Main Rotor

```
           Number of blades =        3
              Rotor radius =     12.5 ft
               Blade twist =    40.00 degs
             Blade airfoil =    VR-12
     Blade lift curve slope =      5.45
              Blade weight =    170.0 lbs
        Rotational velocity =    61.68 rads/sec
          Blade grip length =      1.2 ft
               Hinge offset =      0.0 ft
  Flapping moment of inertia =    102.5 slug ft^2
   Hub height above waterline =    13.0 ft
         Hub fuselage station =    25.0 ft
  Hub position rt of buttline =    16.1 ft
              Mast incidence =     0.00 deg
```

NOTAR

```
       Height above waterline =      0.0 ft^2
            Fuselage station =      0.0 ft^2
   Position right of buttline =      0.0 ft^2
          NOTAR boom diameter =      0.0 ft^2
          Swirl angle at boom =     0.00 deg
      Maximum thruster force =      0.0 lbs
```

Thrust fuselage station =   0.0 ft^2

Wing

```
                          Area =   181.0 ft^2
                          Span =    32.2 ft
                            CL =     0.30
                           CDo = 0.0170
                      Tip cord =     5.3 ft
                     Root cord =     5.3 ft
        Wing efficiency factor =     0.90
               Zero lift angle =    -4.47 deg
            Angle of incidence =     0.00 deg
              Lift curve slope =     4.87
         Height above waterline =     8.0 ft
              Fuselage station =    24.3 ft
     Position right of buttline =     8.5 ft
           Rotor downwash ratio =     1.00
        Fuselage downwash ratio =     1.00
```

Horizontal tail

```
                          Area =    50.2 ft^2
                          Span =    12.8 ft
                            CL =     0.00
                           CDo = 0.0088
               Zero lift angle =     0.00 deg
            Angle of incidence =     0.00 deg
              Lift curve slope =     4.44
         Height above waterline =     8.6 ft
              Fuselage station =    46.7 ft
     Position right of buttline =     0.0 ft
         Dynamic pressure ratio =     1.00
           Rotor downwash ratio =     1.00
        Fuselage downwash ratio =     0.32
```

Vertical tail

```
                          Area =    50.5 ft^2
                          Span =     7.7 ft
                            CL =     0.00
                           CDo = 0.0071
         Height above waterline =     9.6 ft
              Fuselage station =    47.5 ft
     Position right of buttline =     0.0 ft
               Zero lift angle =     0.00 deg
                   Maximum Cl =     1.00
         Dynamic pressure ratio =     1.00
              Lift curve slope =     3.32
```

70

Rigging

```
      Long cyclic pitch/inch defl =     2.10 deg/in
       Lat cyclic pitch/inch defl =     0.62 deg/in
      Collective pitch/inch defl =     1.60 deg/in
   Tail rotor pitch change/defl =     1.60 deg/unit
       Max deflection of control
            from neutral for NOTAR =    0.00 units
   Displacement of anti-torque
       control until full rudder =     5.00 units
```

*** CALCULATED DATA ***

State Matrices

Longitudinal uncoupled plant (A or F depending on notation)
States are [u w q theta]

```
   -0.0124    -0.0025     1.3086    -32.1662
         0    -0.1909     0.1000      0.4963
    0.0015     0.0003    -0.1534          0
         0          0     1.0000          0
```

Longitudinal uncoupled input matrix (B or G depending on notation)
Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]

```
    0.6264    -0.0711          0          0
         0    -5.3507          0          0
   -0.0734     0.0075          0          0
         0          0          0          0
```

Lateral/directional uncoupled plant (A or F depending on notation)
States are [v p phi r psi]

```
   -0.0124    -1.3853    32.1662    -0.1232          0
   -0.0006    -0.4432          0     0.0006          0
         0     1.0000          0    -0.0154          0
    0.0001     0.0230          0    -0.0195          0
         0          0          0     1.0000          0
```

Lateral/directional uncoupled input matrix (B or G depending
on notation)
Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]

```
         0          0    -0.0063          0
         0          0     0.2571     0.0011
         0          0          0          0
         0          0     0.0021     0.0468
         0          0          0          0
```

71

```
Coupled plant (A or F depending on notation)
States are [u w q theta v p phi r psi]

   Columns 1 through 7
   -0.0124    -0.0025     1.3086   -32.1662          0          0          0
        0     -0.1909     0.1000     0.4963          0          0          0
    0.0015     0.0003    -0.1534          0          0          0          0
        0          0     1.0000          0          0          0          0
        0          0          0          0    -0.0124    -1.3853    32.1662
        0          0          0          0    -0.0006    -0.4432          0
        0          0          0          0          0     1.0000          0
        0          0          0          0     0.0001     0.0230          0
        0          0          0          0          0          0          0
   Columns 8 through 9
        0          0
        0          0
        0          0
        0          0
   -0.1232          0
    0.0006          0
   -0.0154          0
   -0.0195          0
    1.0000          0


Coupled input matrix (B or G depending on notation)
Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]

    0.6264    -0.0711          0          0
         0    -5.3507          0          0
   -0.0734     0.0075          0          0
         0          0          0          0
         0          0    -0.0063          0
         0          0     0.2571     0.0011
         0          0          0          0
         0          0     0.0021     0.0468
         0          0          0          0



                    Eigenvalues


Uncoupled

Longitudinal plant

   Eigenvalue          Damping           Freq. (rad/sec)
   -0.4242               1.0000              0.4242
    0.1295 + 0.3051i    -0.3907              0.3314
    0.1295 - 0.3051i    -0.3907              0.3314
   -0.1915               1.0000              0.1915
```

72

Lateral/Directional plant

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
-0.5170	1.0000	0.5170
0.0312 + 0.1885i	-0.1632	0.1911
0.0312 - 0.1885i	-0.1632	0.1911
-0.0204	1.0000	0.0204

Coupled Plant

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
-0.5170	1.0000	0.5170
-0.4242	1.0000	0.4242
0.1295 + 0.3051i	-0.3907	0.3314
0.1295 - 0.3051i	-0.3907	0.3314
0.0312 + 0.1885i	-0.1632	0.1911
0.0312 - 0.1885i	-0.1632	0.1911
-0.1915	1.0000	0.1915
-0.0204	1.0000	0.0204

*** KEY CONTROL PARAMETERS ***

Designed damping
pitch = -3276.1 ft-lbs/(rad/sec)
roll = -23425.8 ft-lbs/(rad/sec)
yaw = -1293.9 ft-lbs/(rad/sec)

Control Power
pitch = -1568.4 ft-lbs/in
roll = 13571.0 ft-lbs/in
yaw = 3103.1 ft-lbs/in

Cooper Harper Pilot Ratings
damping/moment of inertia
pitch (dM/dq)/Iyy = -0.15 [ft-lbs/(rad/sec)]/(slug ft^2)
roll (dR/dp)/Ixx = -0.44 [ft-lbs/(rad/sec)]/(slug ft^2)
yaw (dN/dr)/Izz = -0.02 [ft-lbs/(rad/sec)]/(slug ft^2)

control power/moment of inertia
pitch (dM/in)/Iyy = -0.07 (ft-lbs/in)/(slug ft^2)
roll (dR/in)/Ixx =   0.26 (ft-lbs/in)/(slug ft^2)
yaw (dN/in)/Izz =   0.05 (ft-lbs/in)/(slug ft^2)

74

APPENDIX D. JANRAD OUTPUT FOR XV-15 DESIGN AIRPLANE MODE

```
                       *** RESULTS ***
                            xv15a

                      *** INPUT DATA ***

                      Flight Conditions

            Forward velocity =     200 kts
                 Temperature =      59 degs F
           Pressure altitude =       0 ft
            Auxiliary thrust =       0 lbs


                          Fuselage

                Gross weight =   13000 lbs
    Equivalent flat plate area =     1.6 ft^2
      Vertical projected area =       0.0 ft^2
     Center of Pressure station =    24.4 ft
    Fuselage alpha @ zero lift =      -8.0 degrees
  Lift curve slope of fuselage =     0.286 1/rad
Moment coefficient @ 0 alpha =   -0.070 (referenced to Aw, cw)
      Moment coeff./alpha slope =     1.145 1/rad
      CG height above waterline =       6.0 ft
   Aircraft CG fuselage station =    24.7 ft
  CG position right of buttline =      0.0 ft
                         Ixx = 51039.0 slug ft^2
                         Iyy = 20364.0 slug ft^2
                         Izz = 67096.0 slug ft^2
                         Ixz = 1075.6 slug ft^2
              Downwash ratio =    1.00


                        Main Rotor

            Number of blades =       3
                Rotor radius =    12.5 ft
                 Blade twist =   40.00 degs
               Blade airfoil =   VR-12
     Blade lift curve slope =     4.95
                Blade weight =   170.0 lbs
        Rotational velocity =   54.14 rads/sec
           Blade grip length =      1.2 ft
                Hinge offset =      0.0 ft
   Flapping spring constant =   225.0 ft-lbs/deg
  Flapping moment of inertia =   102.5 slug ft^2
   Hub height above waterline =      8.3 ft
         Hub fuselage station =    25.0 ft
  Hub position rt of buttline =    16.1 ft
              Mast incidence = -90.00 deg
```

Tail rotor (zero if NOTAR)

```
             Number of blades =      0.0
                 Blade chord =      0.0 ft
                Blade radius =      0.0 ft
            Lift curve slope =      0.00
          Rotational velocity =     0.00 rad/sec
   Flapping moment of inertia =      0.0 slug ft^2
               Delta-3 angle =      0.00 deg
                 Blade twist =      0.00 deg
   Hub height above waterline =      0.0 ft
          Hub fuselage station =     0.0 ft
    Hub position rt of buttline =   0.0 ft
```

Wing

```
                        Area =    181.0 ft^2
                        Span =     32.2 ft
                          CL =      0.30
                         CDo = 0.0317
                    Tip cord =      5.3 ft
                   Root cord =      5.3 ft
       Wing efficiency factor =     0.90
             Zero lift angle =     -3.80 deg
          Angle of incidence =      0.00 deg
           Wing sweep angle =     -6.50 deg
         Wing Dihedral angle =      2.50 deg
            Lift curve slope =      5.50 per radian
  Flaperon (roll) effectiveness =   0.34 per radian
  Wing Moment Coeff @ zero lift =   -0.02
        Rotor downwash ratio =      1.00
     Fuselage downwash ratio =      1.00
    Downwash angle @ zero alpha =   0.05 radians
        Height above waterline =    8.0 ft
  Fuselage station (of wing CP) =   24.3 ft
   CP Position right of buttline =   8.5 ft
```

Horizontal tail

```
                        Area =     50.2 ft^2
                        Span =     12.8 ft
                          CL =      0.00
                         CDo = 0.0092
             Zero lift angle =      0.00 deg
          Angle of incidence =      0.00 deg
            Lift curve slope =      4.4
        Height above waterline =    8.6 ft
            Fuselage station =     46.7 ft
    Position right of buttline =    0.0 ft
      Dynamic pressure ratio =      1.00
        Rotor downwash ratio =      1.00
     Fuselage downwash ratio =      0.32
    H-Tail Span Efficiency (e) =    0.80
  Delta Clh per elevator angle =    2.34 per radian
```

76

Vertical tail

Area =	50.5	ft^2
Span =	7.7	ft
CL =	0.00	
CDo =	0.0071	
Height above waterline =	9.6	ft
Fuselage station =	47.5	ft
Position right of buttline =	0.0	ft
Zero lift angle =	0.00	deg
Maximum Cl =	1.00	
Dynamic pressure ratio =	1.00	
Lift curve slope =	3.32	
Clv change with rudder angle =	0.97	per radian

Rigging

Long cyclic pitch/inch defl =	0.00	deg/in
Lat cyclic pitch/inch defl =	0.00	deg/in
Collective pitch/inch defl =	5.00	deg/in
Tail rotor pitch change/defl =	0.00	deg/unit
Max deflection of control from neutral for NOTAR =	0.09	units
Displacement of anti-torque control until full rudder =	5.00	units
Aileron angle/stick defl =	3.93	deg/in
Coll. pitch/stick defl =	5.00	deg/in
Elevator angle/stick defl =	4.17	deg/in
Rudder angle/stick defl =	8.00	deg/in

*** CALCULATED DATA ***

State Matrices

Longitudinal uncoupled plant (A or F depending on notation)
States are [u w q theta]

```
-0.3989    0.0581    7.7690   -32.1615
-0.1909   -1.2933  334.7497    -0.7399
 0.0180   -0.0390   -1.2862          0
      0         0    1.0000          0
```

Longitudinal uncoupled input matrix (B or G depending on notation)
Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]

```
      0    4.9987         0         0
-2.8690         0         0         0
-1.2509   -0.1939         0         0
      0         0         0         0
```

Lateral/directional uncoupled plant (A or F depending on notation)
States are [v p phi r psi]

```
   -0.2609    -8.4066    32.1615  -333.6687           0
   -0.0081    -0.6450          0     0.2056           0
         0     1.0000          0     0.0230           0
    0.0172     0.0526          0    -0.5560           0
         0          0          0     1.0000           0
```

Lateral/directional uncoupled input matrix (B or G depending
on notation)
Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]

```
         0          0          0    -2.3048
         0          0     0.3648    -0.0592
         0          0          0          0
         0          0     0.0016     0.3156
         0          0          0          0
```

Coupled plant (A or F depending on notation)
States are [u w q theta v p phi r psi]

```
   Columns 1 through 7
   -0.3989     0.0581     7.7690   -32.1615          0          0          0
   -0.1909    -1.2933   334.7497    -0.7399          0          0          0
    0.0180    -0.0390    -1.2862          0          0          0          0
         0          0     1.0000          0          0          0          0
         0          0          0          0    -0.2609    -8.4066    32.1615
         0          0          0          0    -0.0081    -0.6450          0
         0          0          0          0          0     1.0000          0
         0          0          0          0     0.0172     0.0526          0
         0          0          0          0          0          0          0
   Columns 8 through 9
         0          0
         0          0
         0          0
         0          0
  -333.6687          0
    0.2056          0
    0.0230          0
   -0.5560          0
    1.0000          0
```

78

Coupled input matrix (B or G depending on notation)
Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]

```
        0      4.9987        0          0
  -2.8690           0        0          0
  -1.2509     -0.1939        0          0
        0           0        0          0
        0           0        0    -2.3048
        0           0   0.3648    -0.0592
        0           0        0          0
        0           0   0.0016     0.3156
        0           0        0          0
```

Eigenvalues

Uncoupled

Longitudinal plant

Eigenvalue	Damping	Freq. (rad/sec)
-1.2941 + 3.5888i	0.3392	3.8150
-1.2941 - 3.5888i	0.3392	3.8150
-0.1951 + 0.1703i	0.7533	0.2590
-0.1951 - 0.1703i	0.7533	0.2590

Lateral/Directional plant

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
-0.3958 + 2.3767i	0.1643	2.4094
-0.3958 - 2.3767i	0.1643	2.4094
-0.6645	1.0000	0.6645
-0.0058	1.0000	0.0058

Coupled Plant

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
-1.2941 + 3.5888i	0.3392	3.8150
-1.2941 - 3.5888i	0.3392	3.8150
-0.3958 + 2.3767i	0.1643	2.4094
-0.3958 - 2.3767i	0.1643	2.4094
-0.6645	1.0000	0.6645
-0.1951 + 0.1703i	0.7533	0.2590
-0.1951 - 0.1703i	0.7533	0.2590
-0.0058	1.0000	0.0058

```
                      *** KEY CONTROL PARAMETERS ***


                              Designed damping
                  pitch = -26191.7 ft-lbs/(rad/sec)
                   roll = -32975.3 ft-lbs/(rad/sec)
                    yaw = -37524.3 ft-lbs/(rad/sec)


                              Control Power
                  pitch = -25473.6 ft-lbs/in
                   roll = 18621.1 ft-lbs/in
                    yaw = 21178.0 ft-lbs/in


                       Cooper Harper Pilot Ratings
                         damping/moment of inertia
        pitch (dM/dq)/Iyy =  -1.29 [ft-lbs/(rad/sec)]/(slug ft^2)
         roll (dR/dp)/Ixx =  -0.65 [ft-lbs/(rad/sec)]/(slug ft^2)
          yaw (dN/dr)/Izz =  -0.56 [ft-lbs/(rad/sec)]/(slug ft^2)


                       control power/moment of inertia
        pitch (dM/in)/Iyy =  -1.25 (ft-lbs/in)/(slug ft^2)
         roll (dR/in)/Ixx =   0.36 (ft-lbs/in)/(slug ft^2)
          yaw (dN/in)/Izz =   0.32 (ft-lbs/in)/(slug ft^2)
```

80

APPENDIX E. GTRS (XV-15) COMPARISON MODELS

GTRS (XV-15) **Hover** Model

Fcoup15h =

Columns 1 through 4

```
-0.0127    -0.0027     1.3154    -32.1662
-0.0707    -0.1984     0.3676      0.4963
 0.0007     0.0000    -0.2007           0
      0          0     1.0000           0
      0          0          0           0
      0          0          0           0
      0          0          0           0
      0          0          0           0
      0          0          0           0
```

Columns 5 through 9

```
      0          0          0          0         0
      0          0          0          0         0
      0          0          0          0         0
      0          0          0          0         0
-0.0570    -1.2538    32.1662    -0.4870         0
-0.0050    -0.3568          0     0.1159         0
      0     1.0000          0          0         0
 0.0012     0.1511          0    -0.0286         0
      0          0          0     1.0000         0
```

Flong15h =

```
-0.0127    -0.0027     1.3154    -32.1662
-0.0707    -0.1984     0.3676      0.4963
 0.0007     0.0000    -0.2007           0
      0          0     1.0000           0
```

Flat15h =

```
-0.0570    -1.2538    32.1662    -0.4870         0
-0.0050    -0.3568          0     0.1159         0
      0     1.0000          0          0         0
 0.0012     0.1511          0    -0.0286         0
      0          0          0     1.0000         0
```

81

```
Gcoup15h =

     1.3300    -0.0843          0          0
     0.0154    -5.3566          0          0
    -0.1887    -0.0029          0          0
          0          0          0          0
          0          0    -0.0434     0.2446
          0          0     0.2411     0.0232
          0          0          0          0
          0          0    -0.0211     0.1006
          0          0          0          0


Glong15h =

     1.3300    -0.0843
     0.0154    -5.3566
    -0.1887    -0.0029
          0          0


Glat15h =

    -0.0434     0.2446
     0.2411     0.0232
          0          0
    -0.0211     0.1006
          0          0
```

GTRS (XV-15) Hover Model Longitudinal Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.0810 + 0.2352i	-0.3256	0.2487
0.0810 - 0.2352i	-0.3256	0.2487
-0.3733	1.0000	0.3733
-0.2005	1.0000	0.2005

GTRS (XV-15) Hover Model Lateral Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0.1445 + 0.4459i	-0.3083	0.4688
0.1445 - 0.4459i	-0.3083	0.4688
-0.7305	1.0000	0.7305
-0.0008	1.0000	0.0008
0	-1.0000	0

82

GTRS (XV-15) **Airplane** Model

Fcoup15a =

Columns 1 through 4

```
-0.4138     0.0729    -6.5286   -32.1621
-0.1709    -1.2073   325.1683    -0.7124
 0.0215    -0.0372    -2.1913          0
      0          0     1.0000          0
      0          0          0          0
      0          0          0          0
      0          0          0          0
      0          0          0          0
      0          0          0          0
```

Columns 5 through 9

```
      0          0          0          0          0
      0          0          0          0          0
      0          0          0          0          0
      0          0          0          0          0
-0.3744     6.3158    32.1621  -328.3823          0
-0.0131    -0.8073          0    -0.0650          0
      0     1.0000          0          0          0
 0.0096    -0.1881          0    -1.0034          0
      0          0          0     1.0000          0
```

Flong15a =

```
-0.4138     0.0729    -6.5286   -32.1621
-0.1709    -1.2073   325.1683    -0.7124
 0.0215    -0.0372    -2.1913          0
      0          0     1.0000          0
```

Flat15a =

```
-0.3744     6.3158    32.1621  -328.3823          0
-0.0131    -0.8073          0    -0.0650          0
      0     1.0000          0          0          0
 0.0096    -0.1881          0    -1.0034          0
      0          0          0     1.0000          0
```

83

```
Gcoup15a =

  -0.0656     5.1084           0           0
  -3.1791     0.0615           0           0
  -1.4324    -0.2439           0           0
        0          0           0           0
        0          0      0.0041     -2.7109
        0          0      0.3339     -0.0694
        0          0           0           0
        0          0      0.0902      0.3816
        0          0           0           0


Glong15a =

  -0.0656     5.1084
  -3.1791     0.0615
  -1.4324    -0.2439
        0          0


Glat15a =

   0.0041    -2.7109
   0.3339    -0.0694
        0          0
   0.0902     0.3816
        0          0
```

GTRS (XV-15) Airplane Model Longitudinal Roots:

Eigenvalue	Damping	Freq. (rad/sec)
-0.2115 + 0.1576i	0.8018	0.2637
-0.2115 - 0.1576i	0.8018	0.2637
-1.6948 + 3.4555i	0.4403	3.8488
-1.6948 - 3.4555i	0.4403	3.8488

GTRS (XV-15) Airplane Model Lateral Roots:

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
-0.1226	1.0000	0.1226
-0.4989 + 1.7702i	0.2712	1.8392
-0.4989 - 1.7702i	0.2712	1.8392
-1.0649	1.0000	1.0649

APPENDIX F. MFS (V-22) COMPARISON MODELS

$A_{\text{af/rotor (hover)}} =$

Columns 1 through 7

```
 -0.0680        0    -0.0011         0     0.5327        0          0
       0  -0.0802         0    -0.5312         0     0.5320    32.1498
 -0.0074        0    -0.1416         0     0.2861        0          0
  0.0000  -0.0141     0.0000   -0.1981     0.0001     0.1800     0.0000
  0.0031   0.0000     0.0000    0.0006    -0.0517    -0.0006         0
  0.0000  -0.0018     0.0000   -0.0079     0.0000    -0.0693     0.0000
       0        0          0    1.0000         0    -0.0388          0
       0        0          0         0     1.0000         0          0
       0        0          0         0         0     1.0008          0
 -0.0388        0    -0.9992         0         0         0          0
       0  -0.0025     0.0000   -0.0252         0         0          0
       0   0.0025     0.0000    0.0252         0         0          0
       0   0.0018     0.0000    0.1763         0         0          0
       0  -0.0018     0.0000   -0.1763         0         0          0
```

Columns 8 through 14

```
 -32.1493        0    0.0000   -13.1278   -13.1278     5.8399     5.8399
        0        0         0    -5.8393     5.8393   -13.1273    13.1273
   1.2461        0    0.0009    -0.0448    -0.0448     0.0050     0.0050
        0        0    0.0000    -0.2927     0.2891    -0.6216     0.6233
   0.0000        0         0     1.4246     1.4256    -0.6578    -0.6627
        0        0    0.0000    -0.9375     0.9390     0.4212    -0.4220
        0        0         0          0          0          0          0
        0        0         0          0          0          0          0
        0        0         0          0          0          0          0
   0.0002        0         0          0          0          0          0
        0        0    0.0000    -9.6324    -0.0001    -4.7116     0.0000
        0        0    0.0000    -0.0001    -9.6324     0.0000    -4.7116
        0        0    0.0000     4.7116     0.0000    -9.6325     0.0001
        0        0    0.0000     0.0000     4.7116     0.0001    -9.6325
```

Eigenvalue	Damping	Freq. (rad/sec)
0.2970 + 0.6539i	-0.4135	0.7181
0.2970 - 0.6539i	-0.4135	0.7181
0.1933 + 0.4004i	-0.4349	0.4446
0.1933 - 0.4004i	-0.4349	0.4446
0	-1.0000	0
-0.0064	1.0000	0.0064
-0.0835	1.0000	0.0835
-0.1352	1.0000	0.1352
-0.5065	1.0000	0.5065
-0.8681	1.0000	0.8681
-9.6275 + 4.7259i	0.8977	10.7249
-9.6275 - 4.7259i	0.8977	10.7249
-9.6325 + 4.7116i	0.8983	10.7230
-9.6325 - 4.7116i	0.8983	10.7230

$A'_{af/rotor}$ =

Columns 1 through 7

```
-0.0680         0   -0.0011         0    0.5327         0         0
      0   -0.0802         0   -0.5312         0    0.5320   32.1498
-0.0074         0   -0.1416         0    0.2861         0         0
 0.0000   -0.0141    0.0000   -0.1981    0.0001    0.1800    0.0000
 0.0031    0.0000    0.0000    0.0006   -0.0517   -0.0006         0
 0.0000   -0.0018    0.0000   -0.0079    0.0000   -0.0693    0.0000
      0         0         0    1.0000        0   -0.0388         0
      0         0         0         0    1.0000         0         0
      0         0         0         0         0    1.0008         0
```

Columns 8 through 9

```
-32.1493         0
       0         0
  1.2461         0
       0         0
  0.0000         0
       0         0
       0         0
       0         0
       0         0
```

A' =

Columns 1 through 4

```
-0.0680   -0.0011    0.5327   -32.1493
-0.0074   -0.1416    0.2861     1.2461
 0.0031    0.0000   -0.0517     0.0000
      0         0    1.0000          0
      0         0         0          0
 0.0000    0.0000    0.0001          0
      0         0         0          0
 0.0000    0.0000    0.0000          0
      0         0         0          0
```

Columns 5 through 9

```
  .0         0         0         0         0
   0         0         0         0         0
 0.0000    0.0006         0   -0.0006         0
   0         0         0         0         0
-0.0802   -0.5312   32.1498    0.5320         0
-0.0141   -0.1981    0.0000    0.1800         0
   0    1.0000         0   -0.0388         0
-0.0018   -0.0079    0.0000   -0.0693         0
   0         0         0    1.0008         0
```

Flon22h =

```
    -0.0680    -0.0011     0.5327   -32.1493
    -0.0074    -0.1416     0.2861     1.2461
     0.0031     0.0000    -0.0517     0.0000
          0          0     1.0000          0
```

Eigenvalue	Damping	Freq. (rad/sec)
0.1933 + 0.4004i	-0.4349	0.4446
0.1933 - 0.4004i	-0.4349	0.4446
-0.1416	1.0000	0.1416
-0.5065	1.0000	0.5065

Flat22h =

```
    -0.0802    -0.5312    32.1498     0.5320          0
    -0.0141    -0.1981     0.0000     0.1800          0
          0     1.0000          0    -0.0388          0
    -0.0018    -0.0079     0.0000    -0.0693          0
          0          0          0     1.0008          0
```

Eigenvalue	Damping	Freq. (rad/sec)
0.3014 + 0.6583i	-0.4163	0.7241
0.3014 - 0.6583i	-0.4163	0.7241
0	-1.0000	0
-0.0918	1.0000	0.0918
-0.8587	1.0000	0.8587

87

$A_{\text{af/rotor (airplane)}} =$

Columns 1 through 7

```
 -0.2295        0    0.0903        0  -22.4870         0   -0.0002
  0.0000   -0.2596   0.0000   21.8531    0.0001 -335.2607   32.1050
 -0.1338        0   -0.7763        0   331.7240   -0.0019        0
  0.0000   -0.0050   0.0000   -0.5810    0.0024   -0.1267    0.0000
  0.0084    0.0000  -0.0031    0.0013   -1.8625   -0.0007        0
  0.0000    0.0011   0.0000   -0.1859   -0.0011   -0.5897   -0.0004
       0         0        0    1.0000         0    0.0677        0
       0         0        0         0    1.0000         0        0
       0         0        0         0         0    1.0023        0
  0.0676        0   -0.9975        0         0         0        0
 -0.0009    0.0011   0.0102   -0.2324   -1.0733   -0.0130        0
 -0.0009   -0.0011   0.0102    0.2324   -1.0733    0.0131        0
 -0.0004    0.0104  -0.0011    0.0543    0.0095    1.0676        0
 -0.0004   -0.0104  -0.0011   -0.0543    0.0095   -1.0676        0
```

Columns 8 through 14

```
 -32.1006        0   -0.0001   -0.5866   -0.5866   -0.5875   -0.5876
        0        0    0.0000   -7.5950    7.5950   14.8716  -14.8716
  -2.1744        0    0.0009   14.8773   14.8773    7.5884    7.5878
        0        0    0.0000   -1.4544    1.4563   -0.4889    0.4907
        0        0    0.0000   -0.8067   -0.8148   -0.6749   -0.6788
        0        0    0.0000   -0.2567    0.2558    0.2188   -0.2196
        0        0         0         0         0         0         0
        0        0         0         0         0         0         0
        0        0         0         0         0         0         0
  337.5816       0         0         0         0         0         0
        0        0    0.0000   -6.1301   -0.0004   -3.0068   -0.0001
        0        0    0.0000   -0.0003   -6.1301   -0.0001   -3.0068
        0        0    0.0000    3.0015   -0.0002   -6.1374    0.0000
        0        0    0.0000   -0.0002    3.0033    0.0000   -6.1374
```

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
-0.0048	1.0000	0.0048
-0.1050	1.0000	0.1050
-0.1063 + 0.9446i	0.1118	0.9505
-0.1063 - 0.9446i	0.1118	0.9505
-0.1139 + 0.2409i	0.4273	0.2665
-0.1139 - 0.2409i	0.4273	0.2665
-0.7851	1.0000	0.7851
-1.0095 + 1.3974i	0.5856	1.7239
-1.0095 - 1.3974i	0.5856	1.7239
-6.2975 + 3.0325i	0.9010	6.9896
-6.2975 - 3.0325i	0.9010	6.9896
-6.4423 + 3.0118i	0.9059	7.1116
-6.4423 - 3.0118i	0.9059	7.1116

$$A'_{af/rotor\ (airplane)} =$$

Columns 1 through 7

```
-0.2295         0    0.0903         0  -22.4870         0   -0.0002
 0.0000   -0.2596    0.0000   21.8531    0.0001 -335.2607   32.1050
-0.1338         0   -0.7763         0  331.7240   -0.0019         0
 0.0000   -0.0050    0.0000   -0.5810    0.0024   -0.1267    0.0000
 0.0084    0.0000   -0.0031    0.0013   -1.8625   -0.0007         0
 0.0000    0.0011    0.0000   -0.1859   -0.0011   -0.5897   -0.0004
      0         0         0    1.0000         0    0.0677         0
      0         0         0         0    1.0000         0         0
      0         0         0         0         0    1.0023         0
```

Columns 8 through 9

```
-32.1006         0
       0         0
 -2.1744         0
       0         0
       0         0
       0         0
       0         0
       0         0
       0         0
```

$$A'_{(airplane)} =$$

Columns 1 through 4

```
-0.2295    0.0903  -22.4870  -32.1006
-0.1338   -0.7763  331.7240   -2.1744
 0.0084   -0.0031   -1.8625         0
      0         0    1.0000         0
 0.0000    0.0000    0.0001         0
 0.0000    0.0000    0.0024         0
      0         0         0         0
 0.0000    0.0000   -0.0011         0
      0         0         0         0
```

Columns 5 through 9

```
      0         0   -0.0002         0    0
      0         0         0   -0.0019    0
 0.0000    0.0013         0   -0.0007    0
      0         0         0         0    0
-0.2596   21.8531   32.1050 -335.2607    0
-0.0050   -0.5810    0.0000   -0.1267    0
      0    1.0000         0    0.0677    0
 0.0011   -0.1859   -0.0004   -0.5897    0
      0         0         0    1.0023    0
```

89

```
Flon22a =

    -0.2295      0.0903   -22.4870   -32.1006
    -0.1338     -0.7763   331.7240    -2.1744
     0.0084     -0.0031    -1.8625          0
          0           0     1.0000          0


    Eigenvalue              Damping          Freq.  (rad/sec)
    -0.1002 + 0.2719i       0.3457              0.2898
    -0.1002 - 0.2719i       0.3457              0.2898
    -1.3340 + 0.9354i       0.8188              1.6293
    -1.3340 - 0.9354i       0.8188              1.6293



Flat22a =

    -0.2596     21.8531    32.1050  -335.2607         0
    -0.0050     -0.5810     0.0000     -0.1267         0
          0      1.0000          0      0.0677         0
     0.0011     -0.1859    -0.0004     -0.5897         0
          0           0          0      1.0023         0


    Eigenvalue              Damping          Freq.  (rad/sec)
          0                 -1.0000                  0
    -0.1212 + 0.8156i       0.1470              0.8245
    -0.1212 - 0.8156i       0.1470              0.8245
    -0.1354                 1.0000              0.1354
    -1.0526                 1.0000              1.0526
```
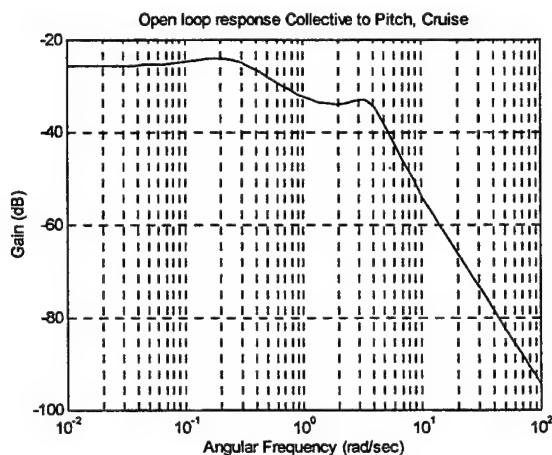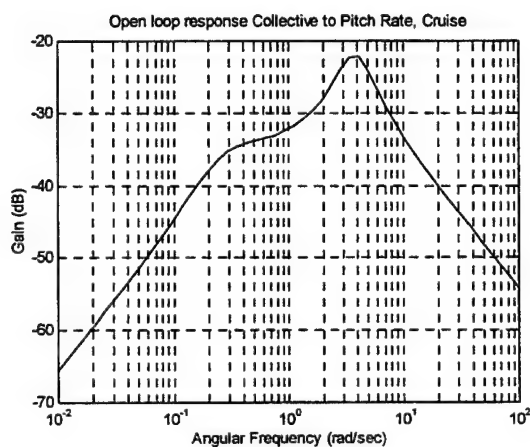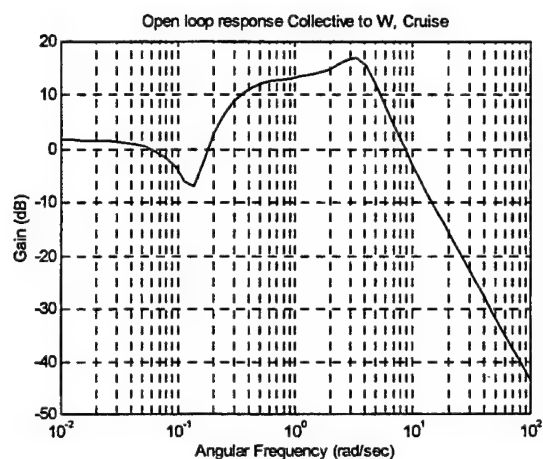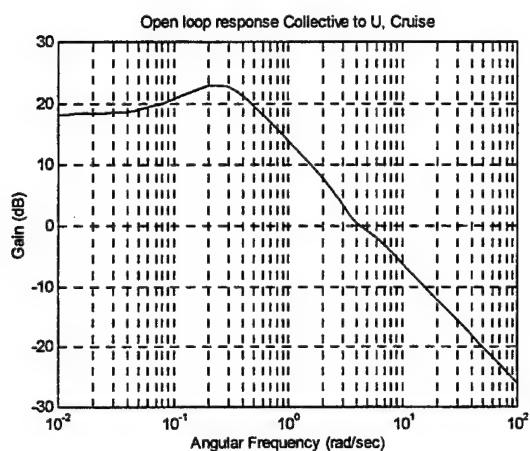
APPENDIX G. JANRAD FREQUENCY RESPONSES

Hover Model

Longitudinal
Response
to
Cyclic Input

Open loop response Longitudinal Cyclic to U, Hover

Open loop response Longitudinal Cyclic to W, Hover

Open loop response Longitudinal Cyclic to Pitch Rate, Hover

Open loop response Longitudinal Cyclic to Theta, Hover

91

Hover Model

Longitudinal
Response
to
Collective Input

Open loop response Collective to U, Hover

Open loop response Collective to W, Hover

Open loop response Collective to Pitch Rate, Hover

Open loop response Collective to Pitch, Hover

Hover Model

Lateral Response
to
Cyclic Input

93

Hover Model

Lateral Response
to
Pedal Input

Open loop response Pedals to Sideslip (v), Hover

Open loop response Pedals to Roll Rate, Hover

Open loop response Pedals to Bank, Hover

Open loop response Pedals to Yaw Rate, Hover

Open loop response Pedals to Yaw, Hover

94

Airplane Model

Longitudinal
Response
to
Cyclic Input

Open loop response Longitudinal Cyclic to U, Cruise

Open loop response Longitudinal Cyclic to W, Cruise

Open loop response Longitudinal Cyclic to Pitch Rate, Cruise

Open loop response Longitudinal Cyclic to Theta, Cruise

Airplane Model

Longitudinal
Response
to
Collective Input

96

Airplane Model

Lateral Response
to
Cyclic Input

Open loop response Lateral Cyclic to Sideslip (v), Cruise

Open loop response Lateral Cyclic to Roll Rate, Cruise

Open loop response Lateral Cyclic to Bank, Cruise

Open loop response Lateral Cyclic to Yaw Rate, Cruise

Open loop response Lateral Cyclic to Yaw, Cruise

97

Airplane Model

Lateral Response
to
Pedal Input

Open loop response Pedals to Sideslip (v), Cruise

Open loop response Pedals to Roll Rate, Cruise

Open loop response Pedals to Bank, Cruise

Open loop response Pedals to Yaw Rate, Cruise

Open loop response Pedals to Yaw, Cruise

APPENDIX H. GTRS MODEL(S) FREQUENCY RESPONSES

Hover Model

Longitudinal
Response
to
Cyclic Input

Open loop response Longitudinal Cyclic to U, Hover

Open loop response Longitudinal Cyclic to W, Hover

Open loop response Longitudinal Cyclic to Pitch Rate, Hover

Open loop response Longitudinal Cyclic to Theta, Hover

Hover Model

Longitudinal
Response
to
Collective Input

Open loop response Collective to U, Hover

Open loop response Collective to W, Hover

Open loop response Collective to Pitch Rate, Hover

Open loop response Collective to Pitch, Hover

Hover Model

Lateral Response
to
Cyclic Input

Open loop response Lateral Cyclic to Sideslip (v), Hover

Open loop response Lateral Cyclic to Roll Rate, Hover

Open loop response Lateral Cyclic to Bank, Hover

Open loop response Lateral Cyclic to Yaw Rate, Hover

Open loop response Lateral Cyclic to Yaw, Hover

101

Hover Model

Lateral Response
to
Pedal Input

Open loop response Pedals to Sideslip (v), Hover

Open loop response Pedals to Roll Rate, Hover

Open loop response Pedals to Bank, Hover

Open loop response Pedals to Yaw Rate, Hover

Open loop response Pedals to Yaw, Hover

102

Airplane Model

Longitudinal
Response
to
Cyclic Input

Open loop response Longitudinal Cyclic to U, Cruise

Open loop response Longitudinal Cyclic to W, Cruise

Open loop response Longitudinal Cyclic to Pitch Rate, Cruise

Open loop response Longitudinal Cyclic to Theta, Cruise

103

Airplane Model

Longitudinal
Response
to
Collective Input

Open loop response Collective to U, Cruise

Open loop response Collective to W, Cruise

Open loop response Collective to Pitch Rate, Cruise

Open loop response Collective to Pitch, Cruise

104

Airplane Model

Lateral Response
to
Cyclic Input

Open loop response Lateral Cyclic to Sideslip (v), Cruise

Open loop response Lateral Cyclic to Roll Rate, Cruise

Open loop response Lateral Cyclic to Bank, Cruise

Open loop response Lateral Cyclic to Yaw Rate, Cruise

Open loop response Lateral Cyclic to Yaw, Cruise

105

Airplane Model

Lateral Response
to
Pedal Input

Open loop response Pedals to Sideslip (v), Cruise

Open loop response Pedals to Roll Rate, Cruise

Open loop response Pedals to Bank, Cruise

Open loop response Pedals to Yaw Rate, Cruise

Open loop response Pedals to Yaw, Cruise

APPENDIX I. JANRAD TIME RESPONSES
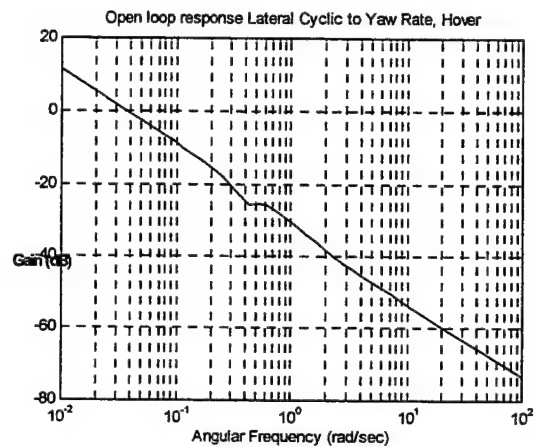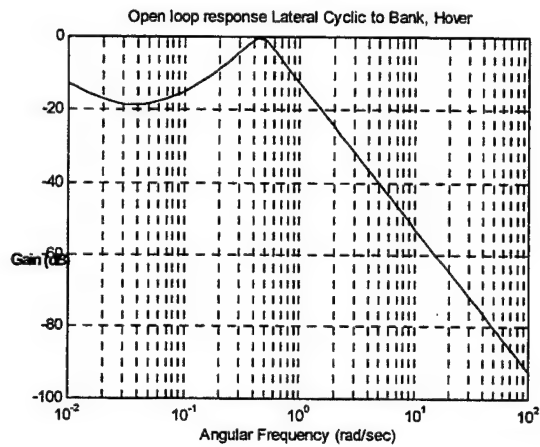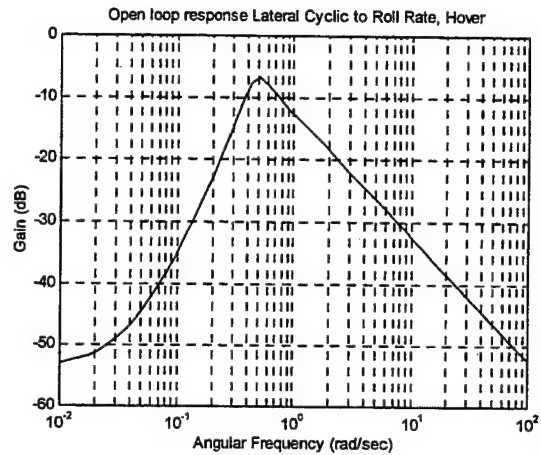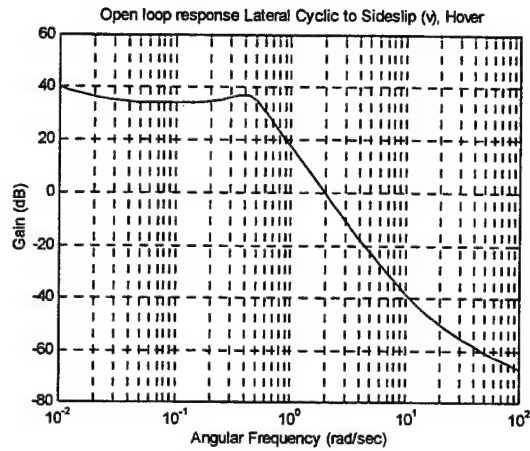
Hover Model

Longitudinal
Response
to
Cyclic Input

107

Hover Model

Longitudinal
Response
to
Collective Input

U Response to Collective step , Hover

W Response to Collective step , Hover

Pitch Rate Response to Collective step, Hover

Pitch Angle Response to Collective step, Hover

Hover Model

Lateral Response
to
Cyclic Input

V Response to Lateral Cyclic Pulse , Hover

Roll Rate Response to Lateral Cyclic Pulse , Hover

Roll Angle Response to Lateral Cyclic Pulse , Hover

Yaw Rate Response to Lateral Cyclic Pulse , Hover

Yaw Angle Response to Lateral Cyclic Pulse , Hover

Hover Model

Lateral Response
to
Pedal Input

V Response to Directional Pedal Doublet , Hover

Roll Rate Response to Directional Pedal Doublet , Hover

Roll Angle Response to Directional Pedal Doublet , Hover

Yaw Rate Response to Directional Pedal Doublet , Hover

Yaw Angle Response to Directional Pedal Doublet , Hover

110

Airplane Model

Longitudinal
Response
to
Cyclic Input

U Response to Longitudinal Cyclic Pulse , Hover

W Response to Longitudinal Cyclic Pulse , Cruise

Pitch Rate Response to Longitudinal Cyclic Pulse , Cruise

Pitch Angle Response to Longitudinal Cyclic Pulse , Cruise

Airplane Model

Longitudinal
Response
to
Collective Input

U Response to Collective step , Cruise

W Response to Collective step , Cruise

Pitch Rate Response to Longitudinal Cyclic Pulse , Cruise

Pitch Angle Response to Collective step, Cruise

Airplane Model

Lateral Response
to
Cyclic Input

V Response to Lateral Cyclic Pulse , Cruise

Roll Rate Response to Lateral Cyclic Pulse , Cruise

Roll Angle Response to Lateral Cyclic Pulse , Cruise

Yaw Rate Response to Lateral Cyclic Pulse , Cruise

Yaw Angle Response to Lateral Cyclic Pulse , Cruise

113

Airplane Model

Lateral Response
to
Pedal Input

V Response to Directional Pedal Doublet , Cruise

Roll Rate Response to Directional Pedal Doublet , Cruise

Roll Angle Response to Directional Pedal Doublet , Cruise

Yaw Rate Response to Directional Pedal Doublet , Cruise

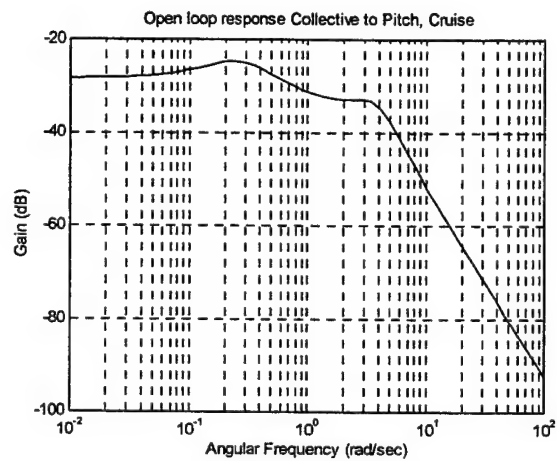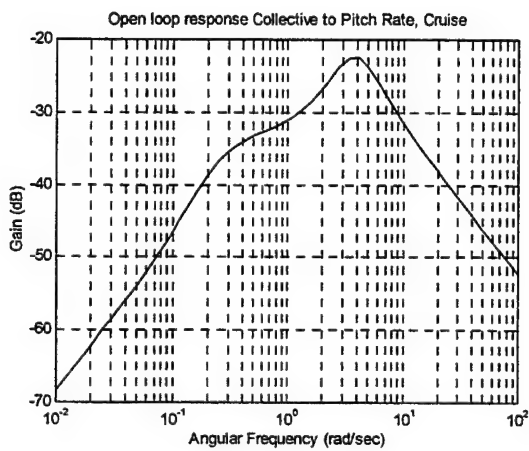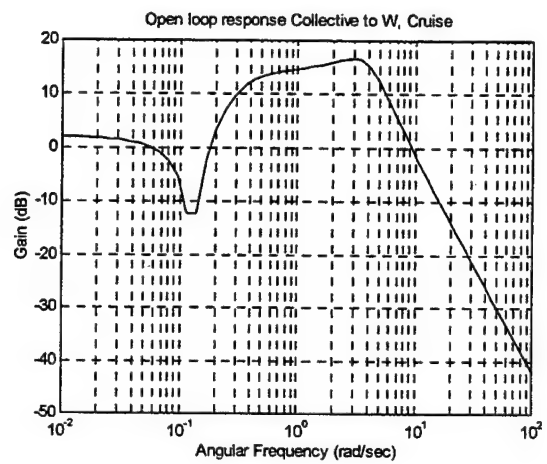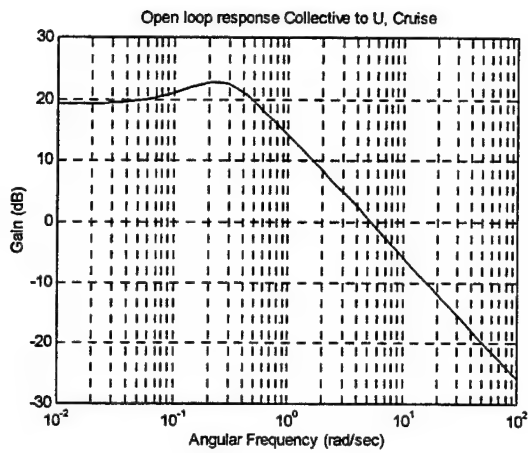Yaw Angle Response to Directional Pedal Doublet , Cruise

114

APPENDIX J. GTRS TIME RESPONSES

Hover Model

Longitudinal
Response
to
Cyclic Input

115

Hover Model

Longitudinal
Response
to
Collective Input

U Response to Collective step , Hover

W Response to Collective step , Hover

Pitch Rate Response to Collective step, Hover

Pitch Angle Response to Collective step, Hover

Hover Model

Lateral Response
to
Cyclic Input

117

Hover Model

Lateral Response
to
Pedal Input

V Response to Directional Pedal Doublet, Hover

Roll Rate Response to Directional Pedal Doublet, Hover

Roll Angle Response to Directional Pedal Doublet, Hover

Yaw Rate Response to Directional Pedal Doublet, Hover

Yaw Angle Response to Directional Pedal Doublet, Hover
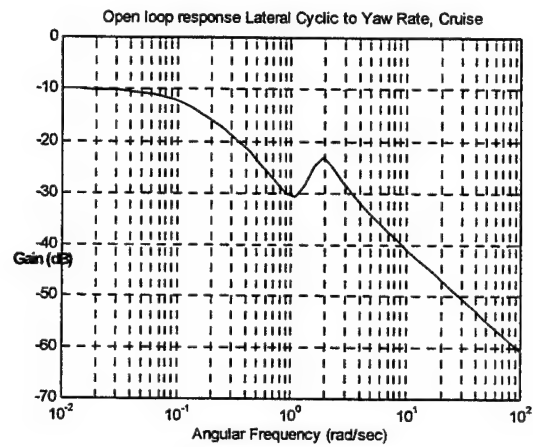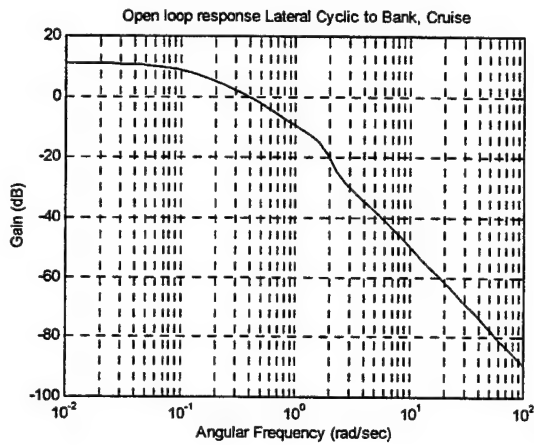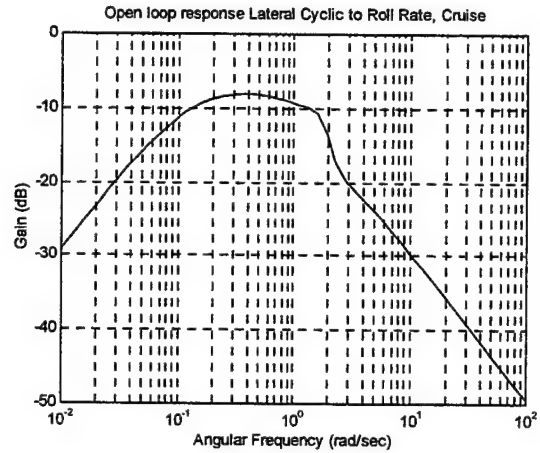
118
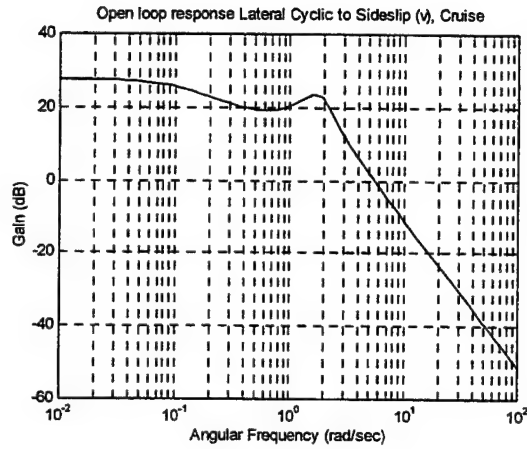
Airplane Model

Longitudinal
Response
to
Cyclic Input

U Response to Longitudinal Cyclic Pulse , Cruise

W Response to Longitudinal Cyclic Pulse , Cruise

Pitch Rate Response to Longitudinal Cyclic Pulse , Cruise

Pitch Angle Response to Longitudinal Cyclic Pulse , Cruise

119

Airplane Model

Longitudinal
Response
to
Collective Input

U Response to Collective step , Cruise

W Response to Collective step , Cruise

Pitch Rate Response to Collective step, Cruise

Pitch Angle Response to Collective step, Cruise

Airplane Model

Lateral Response
to
Cyclic Input

V Response to Lateral Cyclic Pulse , Cruise

Roll Rate Response to Lateral Cyclic Pulse , Cruise

Roll Angle Response to Lateral Cyclic Pulse , Cruise

Yaw Rate Response to Lateral Cyclic Pulse , Cruise

Yaw Angle Response to Lateral Cyclic Pulse , Cruise

Airplane Model

Lateral Response
to
Pedal Input

V Response to Directional Pedal Doublet , Cruise

Roll Rate Response to Directional Pedal Doublet , Cruise

Roll Angle Response to Directional Pedal Doublet , Cruise

Yaw Rate Response to Directional Pedal Doublet , Cruise

Yaw Angle Response to Directional Pedal Doublet , Cruise

APPENDIX K. APPLICABLE EXCERPTS FROM REFERENCE 4

```
***************************************************
***************************************************
    XV-15 TILTROTOR DIGITAL FLIGHT SIMULATION
    GW = 13000 LBS, CG = 299.9 IN, 200 FT.
    BM = 0.0 DEG, RPM = 589.0, FLAPS 40/25, GEAR UP
    DATA FOR GARY KLEIN, NPGS
    TIME: 15:16:07   DATE:  9NOV'95
***************************************************
***************************************************
```

Hover Mode

```
***********    AIRCRAFT TRIM FLIGHT CONDITIONS    ***********

*********************************************************************
*** VT =  0.01 KTS *** NACELLE INCIDENCE = 90.0 DEG *** HELICOPTER ***
*** GW = 13000.0    RPM = 589.00   SLCG = 299.90  WLCG =  81.65 ***
***   MAST ANGLE =   0.00 DEG  *****  FLAP SETTING =   40/25 DEG ***
*********************************************************************
```

```
                       ----- EARTH REFERENCE -----
                        UE          VE          WE
          RATE   (FPS)  0.1000E-01  0.0000E+00  0.0000E+00
          ACCEL  (FPS2) -.9187E-05  -.3777E-07  0.4592E-04
```

```
                     -------------- BODY REFERENCE ---------------
              UB          VB          WB          P          Q          R
RATE  (FT OR DEG/SEC) 0.9999E-02  0.0000E+00  -.1542E-03  0.0000E+00  0.0000E+00  0.0000E+00
ACCEL(FT OR DEG/SEC2)                                     0.1060E-04  -.9961E-03  0.1972E-06
```

```
----- ATMOSPHERIC CONDITIONS -----  ----- FLIGHT PATH CONDITIONS -----  ---- INERTIAS -----  ----- CENTER OF GRAVITY -----
ALTITUDE     (FT)     200.00  CAL. AIRSPEED  (KNOTS)    0.00        (SLUGS-FT2)                    (IN)
DENSITY ALT. (FT)       1.00  RATE OF DECENT (FPS)      0.00
PRESSURE ALT.(FT)       1.00  G-LEVEL        (G S)      1.00   IXX    0.5280E+05      S.L.       299.9
AMBIENT TEMP (DEG-R)  288.16  DYN PRES (SLUGS/FT-SEC2)  0.00   IXZ    1234.           B.L.    0.0000E+00
OUTSIDE AIR TEMP (DEG-C) 15.00  ANGLE OF ATTACK (DEG)  -0.884  IYY    0.2136E+05      W.L.       81.65
AIR DENSITY (SLUGS/FT3) 0.2377E-02  FLIGHT PATH ANGLE(DEG)  0.000  IZZ    0.6634E+05
                              SIDESLIP ANGLE   (DEG)    0.000
THETES (ND)          0.99999  ROLL ANGLE      (DEG)     0.000
DELSTD (ND)          0.99996  PITCH ANGLE     (DEG)    -0.884
SIGMA PRIME (ND)     0.99997  YAW ANGLE       (DEG)     0.000
```

```
----- CONTROL DISPLACEMENTS -----    --------------- POWER/TORQUE --------------- ---------- BLADE ----------
       (+)     (IN)    (PER)          POWER     TORQUE    GOV. PITCH  TIP SPEED    TIP MACH NO.
COLL  (UP)   6.8939   68.94           (SHP)    (FT-LBS)    (DEG)      (FPS)          (ND)
LONG  (FWD)  4.4271   46.12   LEFT    982.9     8765.      0.9774     771.0         0.6906
LAT   (RT)   4.8000   50.00   RIGHT   982.9     8765.      0.9774     771.0         0.6906
PEDAL (RT)   2.5000   50.00   ENGINE  1067.
```

```
----- SWASH PLATE ANGLE (DEG) ----              ---------- ROTOR -------------
        LEFT ROTOR  RIGHT ROTOR  ------------- FLAPPING ------------  ---- FORCES - MAST AXIS ------  -- JET --
THETA0   45.5864     45.5864         A0       LONG      LAT      THRUST    H-FORCE   Y-FORCE   THRUST
B1       -0.7830     -0.7830        (DEG)     (DEG)     (DEG)     (LBS)     (LBS)     (LBS)     (LBS)
A1        0.0000      0.0000   LEFT  2.7964   0.7619   0.1714   7393.43   100.25    24.20     98.56
                               RIGHT 2.7964   0.7619   0.1714   7393.43   100.25    24.20     98.56
```

```
------- SURFACE POSITIONS -------    PROP     ADVANCE   INDUCED   INFLOW   COEFF    COEFF    COEFF
            (DEG)                    EFFIC.   RATIO     VELOCITY  RATIO    POWER    THRUST   DRAG
                                     (ND)     (ND)      (FPS)     (ND)     (ND)     (ND)     (ND)
     ELEVATOR    -1.7655    LEFT   0.0000   0.0000    64.1548   0.0832   0.001011  0.010660  0.010923
     AILERON      0.0000    RIGHT  0.0000   0.0000    64.1548   0.0832   0.001011  0.010660  0.010923
     RUDDER       0.0000
                                     CDRISE   CTEL      CDALPHA   CDLIM    CDMACH   CDFACT
                                     (ND)     (ND)      (ND)      (ND)     (ND)     (ND)
                             LEFT  -0.02234  0.01602   0.01000   0.81000  0.35000  0.20000
                             RIGHT -0.02234  0.01602
```

```
TIME USED FOR THIS TRIM =  0.2565 MIN
```

124

ROTOR DERIVATIVE MATRIX

	U	V	W	P	Q	R	THETO	B1	OMEGA
FAFLPL	0.8193E-02	-.5770E-01	0.1446E-02	-2.349	-3.869	0.1999	0.2054E-02	-.9730	-.5287E-03
FAFLPR	0.8193E-02	0.5770E-01	0.1446E-02	2.349	-3.869	-.1999	0.2054E-02	-.9730	-.5287E-03
LTFLPL	0.8762E-01	0.1490E-01	-.1785E-03	3.872	-2.326	1.447	0.9834E-03	-.2178	0.1102E-02
LTFLPR	0.8762E-01	-.1490E-01	-.1785E-03	-3.872	-2.326	-1.447	0.9834E-03	-.2178	0.1102E-02
HL	2.290	-2.531	0.5367	-316.2	-265.3	42.06	9.471	-128.0	3.170
HR	2.290	2.531	0.5367	316.2	-265.3	-42.06	9.471	-128.0	3.170
YL	4.340	3.070	0.5443E-01	264.2	-307.0	75.07	2.226	-30.85	0.8507
YR	4.340	-3.070	0.5443E-01	-264.2	-307.0	-75.07	2.226	-30.85	0.8507
TL	12.71	-6.437	36.19	-627.4	-7.656	255.3	675.7	-2.248	239.7
TR	12.71	6.467	36.19	626.4	-7.671	-255.2	675.7	-2.248	239.7
HPL	1.803	-.3540E-02	0.2832	-11.13	26.58	50.69	134.9	-.5368	47.81
HPR	1.803	0.6866E-02	0.2832	11.02	26.58	-50.68	134.9	-.5368	47.81
XROT	-4.579	0.5951E-03	-1.073	-.2136E-01	530.6	0.1526E-02	-18.94	255.9	-6.339
YROT	0.0000E+00	-5.914	0.0000E+00	-506.4	-.1526E-03	-159.0	0.0000E+00	0.0000E+00	0.0000E+00
ZROT	-25.25	-.3027E-01	-72.37	0.9473	4.609	-.8789E-01	-1351.	3.420	-479.3
LROT	0.0000E+00	-237.1	0.0000E+00	-.2447E+05	0.2344	6577.	0.0000E+00	0.0000E+00	0.0000E+00
MROT	6.625	-.6042E-02	6.581	0.1868	-4223.	-.1587E-01	58.21	-1919.	23.32
NROT	0.0000E+00	82.37	0.0000E+00	0.1045E+05	-.3906E-01	-1963.	0.0000E+00	0.0000E+00	0.0000E+00
QREQL	16.08	-.3223E-01	2.525	-99.26	234.5	309.7	1202.	-4.786	284.2
QREQR	16.08	0.6152E-01	2.525	98.27	234.5	-309.6	1202.	-4.786	284.2
QLPT	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	-.3052E-04	-.4578E-04	-4.610
QRPT	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	-.3052E-04	-.4578E-04	-4.610

TIME USED IN STABDV =   0.0353 MIN

```
*********************************************************************
*** VT =  0.01 KTS *** NACELLE INCIDENCE = 90.0 DEG  *** HELICOPTER  ***
*** GW = 13000.0     RPM =  589.00   SLCG =  299.90   WLCG =   81.65 ***
***    MAST ANGLE =   0.00 DEG  *****   FLAP SETTING =   40/25 DEG  ***
*********************************************************************
```

		XBODY	YBODY	ZBODY	LBODY	MBODY	NBODY	TORQUE
				UNITS = (LBS OR FT-LB) / (FT/SEC OR RAD/SEC)				
P	TOTAL	0.0000	-506.5822	0.0000	-19022.5000	0.0000	10464.4619	0.0000
	ROTOR	0.0000	-506.4084	0.0000	-24515.7813	0.0000	10460.5078	0.0000
	AIRFRM	0.0000	-0.1739	0.0000	5493.2813	0.0000	3.9534	0.0000
Q	TOTAL	531.5083	0.0000	108.1445	0.0000	-4286.1157	0.0000	548.3203
	ROTOR	531.9072	0.0000	129.5898	0.0000	-4232.6318	0.0000	548.3203
	AIRFRM	-0.3989	0.0000	-21.4453	0.0000	-53.4839	0.0000	0.0000
R	TOTAL	0.0000	-156.3916	0.0000	6152.9321	0.0000	-2037.0656	0.0000
	ROTOR	0.0000	-159.0781	0.0000	6632.9688	0.0000	-1974.2578	0.0000
	AIRFRM	0.0000	2.6865	0.0000	-480.0366	0.0000	-62.8077	0.0000
U	TOTAL	-5.1310	0.0000	-28.5703	-0.0032	15.3265	0.0000	32.3809
	ROTOR	-4.6391	0.0000	-29.3962	-0.0020	6.8644	0.0000	32.3809
	AIRFRM	-0.4919	0.0000	0.8261	-0.0012	8.4621	0.0000	0.0000
V	TOTAL	0.0000	-23.0230	-0.0002	-265.9271	0.0000	85.8339	0.0000
	ROTOR	0.0000	-5.9156	-0.0002	-237.5410	0.0000	82.5278	0.0000
	AIRFRM	0.0000	-17.1073	-0.0001	-28.3861	0.0000	3.3061	0.0000
W	TOTAL	-1.0889	0.0000	-80.1758	0.0000	-0.4589	0.0000	6.5591
	ROTOR	-1.1251	0.0000	-77.4233	0.0000	6.8073	0.0000	6.5591
	AIRFRM	0.0362	0.0000	-2.7526	0.0000	-7.2662	0.0000	0.0000
RPM	TOTAL	-6.3383	0.0000	-423.1283	0.0000	-11.5542	0.0000	829.6618
	ROTOR	-6.3383	0.0000	-479.3777	0.0000	23.3056	0.0000	829.6618
	AIRFRM	0.0000	0.0000	56.2493	0.0000	-34.8598	0.0000	0.0000

CONTROL DERIVATIVE MATRIX

		XBODY	YBODY	ZBODY	LBODY	MBODY	NBODY	TORQUE
				UNITS = (LBS OR FT-LB) / (INCH OR DEG(XTHEG))				
XCOL	TOTAL	-34.0743	0.0000	-2164.6328	0.0000	-60.8882	0.0000	716.4961
	ROTOR	-34.0739	0.0000	-2432.2295	0.0000	104.6219	0.0000	716.4961
	AIRFRM	-0.0004	0.0000	267.5968	0.0000	-165.5101	0.0000	0.0000
XTHEG	TOTAL	-18.5349	0.0000	-1202.7549	0.0000	-36.8198	0.0000	2405.6982
	ROTOR	-18.5347	0.0000	-1351.4336	0.0000	55.1389	0.0000	2405.6982
	AIRFRM	-0.0002	0.0000	148.6782	0.0000	-91.9587	0.0000	0.0000
XLN	TOTAL	537.4612	0.0000	6.2334	0.0000	-4030.1555	0.0000	-20.1113
	ROTOR	537.4612	0.0000	7.1846	0.0000	-4030.7622	0.0000	-20.1113
	AIRFRM	0.0000	0.0000	-0.9517	0.0000	0.6068	0.0000	0.0000
XLT	TOTAL	0.0000	-17.5253	0.0000	12753.1475	0.0000	-1694.2166	0.0000
	ROTOR	0.0000	-17.5253	0.0000	13526.7969	0.0000	-1694.2266	0.0000
	AIRFRM	0.0000	0.0000	0.0000	-773.6499	0.0000	0.0100	0.0000
XPD	TOTAL	0.0000	98.8235	0.0000	1099.7905	0.0000	6643.4497	0.0000
	ROTOR	0.0000	98.8235	0.0000	1093.8125	0.0000	6643.4492	0.0000
	AIRFRM	0.0000	0.0000	0.0000	5.9780	0.0000	0.0005	0.0000

TIME USED IN HANDLE = 0.3572 MIN

```
************     FORCE AND MOMENT SUMMARY (BODY AXIS)     ************

***************************************************************************
*** VT =  0.01 KTS *** NACELLE INCIDENCE = 90.0 DEG  ***  HELICOPTER  ***
*** GW = 13000.0     RPM =  589.00   SLCG =  299.90   WLCG =   81.65 ***
***    MAST ANGLE =   0.00 DEG  *****   FLAP SETTING =    40/25 DEG  ***
***************************************************************************
```

	X-FORCE LBS	Y-FORCE LBS	Z-FORCE LBS	ROLL FT-LBS	PITCH FT-LBS	YAW FT-LBS
FUSELAGE	0.000	0.000	0.000	0.000	0.000	0.000
WING						
LEFT WING	0.014	0.000	841.089	-9058.984	-537.765	0.148
RIGHT WING	0.014	0.000	841.089	9058.984	-537.765	-0.148
FREESTREAM	0.000	0.000	0.000	0.000	0.000	0.000
TOTAL	0.028	0.000	1682.177	0.000	-1075.530	0.000
ENGINE PYLONS	-0.028	0.000	290.515	0.000	2.464	0.000
HORIZ. STAB.	0.000	0.000	0.000	0.000	0.000	0.000
VERT. STAB.						
FIN NO. 1	0.000	0.000	0.000	0.000	0.000	0.000
FIN NO. 2	0.000	0.000	0.000	0.000	0.000	0.000
TOTAL	0.000	0.000	0.000	0.000	0.000	0.000
LANDING GEAR						
MAIN AERO.	0.000	0.000	0.000	0.000	0.000	0.000
NOSE AERO.	0.000	0.000	0.000	0.000	0.000	0.000
DYNAMIC	0.000	0.000	0.000	0.000	0.000	0.000
TOTAL	0.000	0.000	0.000	0.000	0.000	0.000
JET THRUST						
LEFT ENG	0.000	0.000	-98.568	1585.303	-0.821	0.000
RIGHT ENG	0.000	0.000	-98.568	-1585.303	-0.821	0.000
TOTAL	0.000	0.000	-197.136	0.000	-1.643	0.000
GROUND EFFECT						
FUSE-HOR STAB	0.000	0.000	0.000	0.000	0.000	0.000
TOTAL AIRFRAME	-0.001	0.000	1775.557	0.000	-1074.709	0.000
ROTOR						
LEFT ROTOR	-100.239	-153.233	-7392.100	118402.734	537.118	-10385.341
RIGHT ROTOR	-100.239	153.233	-7392.100	-118402.734	537.118	10385.341
HUB SPINNER	-0.002	0.000	9.783	0.000	0.091	0.000
TOTAL ROTOR	-200.479	0.000	-14774.416	0.000	1074.327	0.000
TOTAL AIRCRAFT (BODY AXIS)	-200.480	0.000	-12998.859	0.000	-0.382	0.000
TOTAL AIRCRAFT (INERTIAL AXIS)	-0.010	0.000	-0.405	0.000	-0.382	0.000

127

Airplane Mode

```
************    AIRCRAFT TRIM FLIGHT CONDITIONS    ***********

*********************************************************************
*** VT =200.00 KTS *** NACELLE INCIDENCE =  0.0 DEG ***   AIRPLANE   ***
*** GW = 13000.0    RPM =  517.00   SLCG = 296.34   WLCG =   72.42 ***
***   MAST ANGLE =  90.00 DEG  *****   FLAP SETTING =     0/0  DEG ***
*********************************************************************
```

```
                          ----- EARTH REFERENCE -----
                            UE          VE          WE
              RATE  (FPS)   337.6      0.0000E+00  0.0000E+00
              ACCEL (FPS2)  0.5303E-04 -.1833E-05  -.1155
```

```
                     --------------- BODY REFERENCE ---------------
                     UB          VB          WB        P           Q           R
RATE  (FT OR DEG/SEC)   337.5    0.3271E-06  7.473   0.0000E+00  0.0000E+00  0.0000E+00
ACCEL(FT OR DEG/SEC2)                                -.1247E-04  0.6688E-02  0.2419E-04
```

----- ATMOSPHERIC CONDITIONS -----	----- FLIGHT PATH CONDITIONS -----	----- INERTIAS -----	----- CENTER OF GRAVITY -----
ALTITUDE (FT) 200.00	CAL. AIRSPEED (KNOTS) 200.00	(SLUGS-FT2)	(IN)
DENSITY ALT. (FT) 1.00	RATE OF DECENT (FPS) 0.00		
PRESSURE ALT.(FT) 1.00	G-LEVEL (G S) 1.00	IXX 0.5104E+05	S.L. 296.3
AMBIENT TEMP (DEG-R) 288.16	DYN PRES (SLUGS/FT-SEC2) 135.42	IXZ 1076.	B.L. 0.0000E+00
OUTSIDE AIR TEMP (DEG-C) 15.00	ANGLE OF ATTACK (DEG) 1.269	IYY 0.2036E+05	W.L. 72.42
AIR DENSITY (SLUGS/FT3) 0.2377E-02	FLIGHT PATH ANGLE(DEG) 0.000	IZZ 0.6710E+05	
	SIDESLIP ANGLE (DEG) 0.000		
THETES (ND) 0.99999	ROLL ANGLE (DEG) 0.000		
DELSTD (ND) 0.99996	PITCH ANGLE (DEG) 1.269		
SIGMA PRIME (ND) 0.99997	YAW ANGLE (DEG) 0.000		

```
----- CONTROL DISPLACEMENTS -----    --------------- POWER/TORQUE --------------- ---------- BLADE ----------
          (+)      (IN)    (PER)           POWER    TORQUE    GOV. PITCH  TIP SPEED    TIP MACH NO.
COLL  (UP)   4.9173   49.17             (SHP)    (FT-LBS)   (DEG)      (FPS)         (ND)
LONG  (FWD)  5.2790   54.99    LEFT     688.0    6989.      11.64      756.3         0.6774
LAT   (RT)   4.8000   50.00    RIGHT    688.0    6989.      11.64      756.3         0.6774
PEDAL (RT)   2.5000   50.00    ENGINE   749.7
```

```
----- SWASH PLATE ANGLE (DEG) ----                    ----------- ROTOR -------------
         LEFT ROTOR   RIGHT ROTOR   ------------- FLAPPING ------------   ---- FORCES - MAST AXIS ------  -- JET --
THETA0   67.4386      67.4386              A0      LONG      LAT      THRUST   H-FORCE   Y-FORCE   THRUST
B1        1.4835       1.4835             (DEG)    (DEG)     (DEG)    (LBS)    (LBS)     (LBS)     (LBS)
A1        0.2217       0.2217    LEFT     0.8929  -1.5480   -0.1482   888.05   -35.69    -22.98    -26.04
                                 RIGHT    0.8929  -1.5480   -0.1482   888.05   -35.69    -22.98    -26.04
```

```
------- SURFACE POSITIONS -------      PROP      ADVANCE   INDUCED   INFLOW   COEFF     COEFF     COEFF
          (DEG)                        EFFIC.    RATIO     VELOCITY  RATIO    POWER     THRUST    DRAG
                                       (ND)      (ND)      (FPS)     (ND)     (ND)      (ND)      (ND)
     ELEVATOR   2.2679       LEFT      0.7922    0.0110    1.2902    0.5006   0.001046  0.001662  0.011373
     AILERON    0.0000       RIGHT     0.7922    0.0110    1.2902    0.5006   0.001046  0.001662  0.011373
     RUDDER     0.0000
                                       CDRISE    CTEL      CDALPHA   CDLIM    CDMACH    CDFACT
                                       (ND)      (ND)      (ND)      (ND)     (ND)      (ND)
                             LEFT     -0.02632   0.01521   0.01000   0.81000  0.35000   0.20000
                             RIGHT    -0.02632   0.01521
```

```
TIME USED FOR THIS TRIM =   0.3575 MIN
```

ROTOR DERIVATIVE MATRIX

	U	V	W	P	Q	R	THETO	B1	OMEGA
FAFLPL	-.2213E-02	-.2426E-01	0.6323E-01	-1.079	-4.287	-2.483	0.5380E-02	-1.300	0.6643E-02
FAFLPR	-.2213E-02	0.2426E-01	0.6323E-01	1.079	-4.287	2.483	0.5380E-02	-1.300	0.6643E-02
LTFLPL	-.2768E-02	0.6330E-01	0.2388E-01	-.2488	-2.439	4.247	0.2280E-01	-.4153	0.1025E-01
LTFLPR	-.2768E-02	-.6330E-01	0.2388E-01	0.2488	-2.439	-4.247	0.2280E-01	-.4153	0.1025E-01
HL	2.086	0.1795	15.81	-237.2	1109.	-94.65	-32.86	-99.51	-16.68
HR	2.086	-.1795	15.81	237.2	1109.	94.65	-32.86	-99.51	-16.68
YL	0.4989	15.93	-.1425	41.80	-129.9	-1105.	-6.880	-29.73	-2.943
YR	0.4989	-15.93	-.1425	-41.80	-129.9	1105.	-6.880	-29.73	-2.943
TL	-77.17	-1.702	0.7140	-533.2	193.9	-1253.	1010.	4.936	518.3
TR	-77.17	1.699	0.7139	533.0	193.9	1253.	1010.	4.936	518.3
HPL	-43.28	-.7415	-.3207	-307.2	58.75	-694.3	592.4	1.625	310.7
HPR	-43.28	0.7420	-.3207	307.3	58.75	694.3	592.4	1.625	310.7
XROT	-154.3	-.2586E-02	1.428	-.2423	387.8	-.2351	2020.	9.872	1037.
YROT	-.2825E-06	-31.86	-.8476E-06	-83.60	0.1695E-03	2210.	-.4959E-04	-.2575E-04	-.3643E-05
ZROT	-4.172	0.1130E-05	-31.62	-.1921E-02	-2218.	-.9041E-03	65.72	199.0	33.35
LROT	0.2170E-03	-82.54	-.7233E-04	-.1381E+05	0.0000E+00	-.1207E+05	-.1221E-02	0.0000E+00	-.4663E-03
MROT	370.6	0.6003E-02	155.0	0.5714	7585.	0.5497	-4905.	-1326.	-2513.
NROT	0.0000E+00	-214.4	0.7233E-03	-.1732E+05	0.1447E-01	-.3219E+05	-.4883E-03	-.9766E-03	0.0000E+00
QREQL	-439.7	-7.532	-3.258	-2992.	596.8	-7053.	6018.	16.50	3027.
QREQR	-439.7	7.538	-3.258	2992.	596.8	7054.	6018.	16.50	3027.
QLPT	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	-.3052E-04	-.7629E-04	-3.950
QRPT	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	-.3052E-04	-.7629E-04	-3.950

TIME USED IN STABDV =   0.0397 MIN

```
*********************************************************************
*** VT =200.00 KTS *** NACELLE INCIDENCE = 0.0 DEG ***   AIRPLANE   ***
*** GW = 13000.0    RPM = 517.00  SLCG = 296.34  WLCG =   72.42 ***
***   MAST ANGLE = 90.00 DEG  *****  FLAP SETTING =    0/0 DEG ***
*********************************************************************
```

		XBODY	YBODY	ZBODY	LBODY	MBODY	NBODY	TORQUE
			UNITS =	(LBS OR FT-LB)	/	(FT/SEC OR RAD/SEC)		
P	TOTAL	-0.0036	-467.6222	0.0000	-41002.5742	0.0217	-11752.9883	0.0000
	ROTOR	-0.0018	-83.6027	-0.0001	-13805.2432	0.0072	-17328.9219	0.0000
	AIRFRM	-0.0018	-384.0195	0.0000	-27197.3301	0.0145	5575.9351	0.0000
Q	TOTAL	381.6542	-0.0001	-4983.2866	0.0001	-44624.1445	0.0285	1186.1173
	ROTOR	390.7816	-0.0001	-2217.9490	0.0000	7577.8271	0.0289	1186.1173
	AIRFRM	-9.1274	0.0000	-2765.3359	0.0001	-52201.9688	-0.0005	0.0000
R	TOTAL	0.0181	3684.5029	0.0000	-2239.5098	-0.0723	-67257.1719	0.4195
	ROTOR	0.0145	2209.7168	0.0027	-12072.2031	-0.0434	-32195.2461	0.4195
	AIRFRM	0.0036	1474.7860	0.0000	9832.6934	-0.0289	-35061.9219	0.0000
U	TOTAL	-167.2198	0.0000	-69.0474	-0.0001	438.2175	0.0001	-883.4636
	ROTOR	-154.0378	0.0000	-4.1584	0.0000	369.8365	0.0000	-883.4636
	AIRFRM	-13.1820	0.0000	-64.8890	-0.0001	68.3810	0.0001	0.0000
V	TOTAL	0.0000	-151.3120	0.0000	-676.9139	0.0000	655.5299	-0.0001
	ROTOR	0.0000	-31.9718	0.0000	-82.5311	0.0000	-214.7779	-0.0001
	AIRFRM	0.0000	-119.3403	0.0000	-594.3828	0.0000	870.3078	0.0000
W	TOTAL	29.4440	0.0000	-487.8825	0.0000	-758.4738	0.0000	-6.6435
	ROTOR	1.4010	0.0000	-31.6017	0.0000	154.9586	-0.0001	-6.6435
	AIRFRM	28.0431	0.0000	-456.2809	0.0000	-913.4324	0.0000	0.0000
RPM	TOTAL	1059.7207	0.0000	15.9046	0.0015	-2563.4771	0.0000	6299.7041
	ROTOR	1036.6965	0.0000	33.3510	0.0015	-2513.5779	0.0000	6299.7041
	AIRFRM	23.0242	0.0000	-17.4466	-0.0001	-49.8991	0.0000	0.0000

CONTROL DERIVATIVE MATRIX

		XBODY	YBODY	ZBODY	LBODY	MBODY	NBODY	TORQUE
			UNITS =	(LBS OR FT-LB)	/	(INCH OR DEG(XTHEG))		
XCOL	TOTAL	-0.0002	0.0000	0.0000	0.0000	0.0010	0.0000	-3491.0049
	ROTOR	-0.0002	0.0000	0.0000	0.0000	0.0010	0.0000	-3491.0049
	AIRFRM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
XTHEG	TOTAL	2064.3198	0.0000	24.8711	0.0001	-4967.3730	0.0000	12032.2373
	ROTOR	2020.6517	0.0000	58.8772	0.0002	-4872.8916	0.0000	12032.2373
	AIRFRM	43.6681	0.0000	-34.0059	-0.0001	-94.4814	0.0000	0.0000
XLN	TOTAL	-26.5242	0.0000	-1284.6719	0.0000	-29169.6289	0.0000	0.0000
	ROTOR	-0.0002	0.0000	0.0000	0.0000	0.0010	0.0000	0.0000
	AIRFRM	-26.5239	0.0000	-1284.6719	0.0000	-29169.6309	0.0000	0.0000
XLT	TOTAL	-0.0026	1.6652	-0.0020	16946.8770	0.0078	5694.5244	-0.0137
	ROTOR	-0.0023	1.6652	-0.0002	1332.2534	0.0059	3973.5186	-0.0137
	AIRFRM	-0.0002	0.0000	-0.0020	15614.6230	0.0020	1721.0057	0.0000
XPD	TOTAL	-0.0001	-1095.4661	0.0000	-3950.0532	0.0000	25679.7207	0.0000
	ROTOR	0.0000	0.0000	0.0000	0.0005	0.0000	-0.0029	0.0000
	AIRFRM	-0.0001	-1095.4661	0.0000	-3950.0537	0.0000	25679.7227	0.0000

TIME USED IN HANDLE =   0.3718 MIN

130

```
*************************
* LONGITUDINAL PHUGOID *
*************************


    REAL                        -.21033561
    DAMPED NATURAL FREQUENCY    0.15832365
    UNDAMPED FREQUENCY (RPS)    0.26326308
    UNDAMPED FREQUENCY (CPS)    0.41899655E-01
    PERIOD IN SEC               39.685669
    DAMPING                     0.79895598
    TIME TO HALF                3.2954333



*****************************
* LONGITUDINAL SHORT PERIOD *
*****************************


    REAL                        -2.0009155
    DAMPED NATURAL FREQUENCY     3.2784851
    UNDAMPED FREQUENCY (RPS)     3.8408499
    UNDAMPED FREQUENCY (CPS)     0.61129075
    PERIOD IN SEC                1.9164889
    DAMPING                      0.52095646
    TIME TO HALF                 0.34641492



**************
* DUTCH ROLL *
**************


    REAL                        -.50014758
    DAMPED NATURAL FREQUENCY     1.7704489
    UNDAMPED FREQUENCY (RPS)     1.8397382
    UNDAMPED FREQUENCY (CPS)     0.29280367
    PERIOD IN SEC                3.5489192
    DAMPING                      0.27185801
    TIME TO HALF                 1.3858849



*************
* ROLL MODE *
*************
    REAL                        -1.0648769
    TIME TO HALF                0.65091748



***************
* SPIRAL MODE *
***************
    REAL                        -.12006846
    TIME TO HALF                5.7729315
```

*** COUPLED STABILITY ROOTS - SCAS-OFF - FLAPPING ALLOWED TO CHANGE - 6 X 6 MATRIX ***

MASS MATRIX--

```
0.40407E+03  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.40583E+03  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.38693E+02  0.20364E+05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.40407E+03  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.51039E+05 -0.10756E+04  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00 -0.10756E+04  0.67096E+05  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.10000E+01  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.10000E+01
```

DAMPING MATRIX--

```
 0.16722E+03 -0.29444E+02  0.26380E+04 -0.90406E-05  0.36163E-02 -0.18213E-01  0.12997E+05  0.00000E+00
 0.69047E+02  0.48788E+03 -0.13138E+06  0.00000E+00  0.13218E-03  0.00000E+00  0.28780E+03  0.56890E-03
-0.43822E+03  0.75847E+03  0.44624E+05  0.00000E+00 -0.21698E-01  0.72325E-01  0.00000E+00  0.00000E+00
 0.49506E-05 -0.34436E-07  0.60984E-04  0.15131E+03 -0.25520E+04  0.13268E+06  0.12598E-04 -0.12997E+05
 0.74887E-04 -0.36069E-04 -0.50779E-04  0.67691E+03  0.41003E+05  0.22395E+04  0.00000E+00  0.00000E+00
-0.13674E-03  0.45203E-04 -0.28478E-01 -0.65553E+03  0.11753E+05  0.67257E+05  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00 -0.10000E+01  0.00000E+00  0.00000E+00  0.43772E-07  0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00 -0.96928E-09  0.00000E+00 -0.10000E+01 -0.22144E-01  0.00000E+00  0.00000E+00
```

-STIFFNESS MATRIX--

```
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
```

NO. OF EIGENVALUES CALCULATED   16

| EIGENVALUES, (COMPLEX-VALUED) | | FREQUENCIES | | | DAMPING RATIO | PERIOD | TIME TO |
REAL	IMAG	DAMPED RPS	UNDAMPED RPS	UNDAMPED CPS		SEC	HALF AMPL SEC
-0.2862294E-16	0.0000000E+00	0.0000000E+00	0.2862294E-16	0.4555482E-17	1.000000	0.9999900E+38	0.2421649E+17
0.3238712E-10	0.0000000E+00	0.0000000E+00	0.3238712E-10	0.5154569E-11	-1.000000	0.9999900E+38	0.2140194E+11
0.2059005E-09	0.0000000E+00	0.0000000E+00	0.2059005E-09	0.3277008E-10	-1.000000	0.9999900E+38	0.3366417E+10
-0.1698214E-08	0.0000000E+00	0.0000000E+00	0.1698214E-08	0.2702792E-09	1.000000	0.9999900E+38	0.4081623E+09
0.3316228E-08	0.0000000E+00	0.0000000E+00	0.3316228E-08	0.5277941E-09	-1.000000	0.9999900E+38	0.2090167E+09
0.2548163E-07	0.0000000E+00	0.0000000E+00	0.2548163E-07	0.4055527E-08	-1.000000	0.9999900E+38	0.2720183E+08
0.2548163E-07	0.0000000E+00	0.0000000E+00	0.2548163E-07	0.4055527E-08	-1.000000	0.9999900E+38	0.2720183E+08
-0.5831103E-07	0.0000000E+00	0.0000000E+00	0.5831103E-07	0.9280489E-08	1.000000	0.9999900E+38	0.1187706E+08
-0.1205123	0.0000000E+00	0.0000000E+00	0.1205123	0.1918013E-01	1.000000	0.9999900E+38	5.751670
-1.064887	0.0000000E+00	0.0000000E+00	1.064887	0.1694820	1.000000	0.9999900E+38	0.6509116

-0.2103941	-0.1590430	-0.1590430	0.2637430	0.4197600E-01	0.7977240	-39.50620	3.294517
-0.2103943	0.1590432	0.1590432	0.2637432	0.4197603E-01	0.7977239	39.50616	3.294514
-0.4999204	-1.770403	-1.770403	1.839632	0.2927865	0.2717502	-3.549015	1.386515
-0.4999205	1.770402	1.770402	1.839631	0.2927864	0.2717504	3.549016	1.386514
-2.000854	3.278414	3.278414	3.840757	0.6112755	0.5209531	1.916532	0.3464255
-2.000853	-3.278414	-3.278414	3.840757	0.6112754	0.5209529	-1.916532	0.3464257

TIME REQD FOR THIS PROBLEM   0.024 MIN

*** RPM COUPLED STABILITY ROOTS - SCAS-OFF - FLAPPING ALLOWED TO CHANGE - 7 X 7 MATRIX ***

MASS MATRIX--

```
0.40407E+03  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.40583E+03  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.38693E+02  0.20364E+05  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.40407E+03  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.51039E+05 -0.10756E+04  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00 -0.10756E+04  0.67096E+05  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.82400E+03  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.10000E+01  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.10000E+01
```


DAMPING MATRIX--

```
0.16722E+03 -0.29444E+02  0.26380E+04 -0.90406E-05  0.36163E-02 -0.18213E-01 -0.10597E+04  0.12997E+05  0.00000E+00
0.69047E+02  0.48788E+03 -0.13138E+06  0.00000E+00  0.13218E-03  0.00000E+00 -0.15905E+02  0.28780E+03  0.56890E-03
-0.43822E+03  0.75847E+03  0.44624E+05  0.00000E+00 -0.21698E-01  0.72325E-01  0.25635E+04  0.00000E+00  0.00000E+00
0.49506E-05 -0.34436E-07  0.60984E-04  0.15131E+03 -0.25520E+04  0.13268E+06 -0.18203E-04  0.12598E-04 -0.12997E+05
0.74887E-04 -0.36069E-04 -0.50779E-04  0.67691E+03  0.41003E+05  0.22395E+04 -0.14639E-02  0.00000E+00  0.00000E+00
-0.13674E-03  0.45203E-04 -0.28478E-01 -0.65553E+03  0.11753E+05  0.67257E+05  0.72855E-05  0.00000E+00  0.00000E+00
-0.88346E+03 -0.66435E+01  0.11861E+04 -0.72325E-04  0.00000E+00  0.41949E+00  0.62997E+04  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00 -0.10000E+01  0.00000E+00  0.00000E+00  0.43772E-07  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00 -0.96928E-09  0.00000E+00 -0.10000E+01 -0.22144E-01  0.00000E+00  0.00000E+00  0.00000E+00
```


STIFFNESS MATRIX--

```
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
```


NO. OF EIGENVALUES CALCULATED   18

133

| EIGENVALUES,(COMPLEX-VALUED) | | ---------------FREQUENCIES--------------- | | | DAMPING | PERIOD | TIME TO |
REAL	IMAG	DAMPED RPS	UNDAMPED RPS	UNDAMPED CPS	RATIO	SEC	HALF AMPL SEC
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.9999900E+38	0.9999900E+38	0.9999900E+38
-0.3508324E-15	0.0000000E+00	0.0000000E+00	0.3508324E-15	0.5583670E-16	1.000000	0.9999900E+38	0.1975721E+16
0.5414031E-10	0.0000000E+00	0.0000000E+00	0.5414031E-10	0.8616698E-11	-1.000000	0.9999900E+38	0.1280279E+11
0.5414027E-10	0.0000000E+00	0.0000000E+00	0.5414027E-10	0.8616692E-11	-1.000000	0.9999900E+38	0.1280280E+11
-0.1315212E-07	0.0000000E+00	0.0000000E+00	0.1315212E-07	0.2093225E-08	1.000000	0.9999900E+38	0.5270229E+08
-0.1801357E-07	0.0000000E+00	0.0000000E+00	0.1801357E-07	0.2866948E-08	1.000000	0.9999900E+38	0.3847916E+08
-0.2224268E-07	0.0000000E+00	0.0000000E+00	0.2224268E-07	0.3540032E-08	1.000000	0.9999900E+38	0.3116293E+08
-0.1124087E-06	0.0000000E+00	0.0000000E+00	0.1124087E-06	0.1789041E-07	1.000000	0.9999900E+38	6166309.
0.2519187E-05	0.0000000E+00	0.0000000E+00	0.2519187E-05	0.4009410E-06	-1.000000	0.9999900E+38	275147.1
-0.1205120	0.0000000E+00	0.0000000E+00	0.1205120	0.1918008E-01	1.000000	0.9999900E+38	5.751685
-0.2560959E-01	0.1476254	0.1476254	0.1498303	0.2384623E-01	0.1709239	42.56167	27.06592
-0.2560963E-01	-0.1476256	-0.1476256	0.1498304	0.2384626E-01	0.1709240	-42.56163	27.06588
-1.064887	0.0000000E+00	0.0000000E+00	1.064887	0.1694821	1.000000	0.9999900E+38	0.6509112
-0.4999211	1.770402	1.770402	1.839631	0.2927864	0.2717507	3.549016	1.386513
-0.4999207	-1.770402	-1.770402	1.839632	0.2927865	0.2717504	-3.549016	1.386514
-1.991692	3.273934	3.273934	3.832164	0.6099077	0.5197304	1.919155	0.3480192
-8.033176	0.0000000E+00	0.0000000E+00	8.033176	1.278520	1.000000	0.9999900E+38	0.8628555E-01
-1.991691	-3.273930	-3.273930	3.832160	0.6099072	0.5197306	-1.919157	0.3480194

TIME REQD FOR THIS PROBLEM   0.032 MIN

134

```
*********** FORCE AND MOMENT SUMMARY (BODY AXIS)   ***********

*********************************************************************
*** VT =200.00 KTS *** NACELLE INCIDENCE =  0.0 DEG  ***  AIRPLANE  ***
*** GW = 13000.0    RPM =  517.00   SLCG =  296.34   WLCG =  72.42 ***
***   MAST ANGLE =  90.00 DEG   *****   FLAP SETTING =    0/0  DEG  ***
*********************************************************************
```

	X-FORCE LBS	Y-FORCE LBS	Z-FORCE LBS	ROLL FT-LBS	PITCH FT-LBS	YAW FT-LBS
FUSELAGE	-196.385	0.000	-1139.147	0.000	-5235.144	0.000
WING						
LEFT WING	0.000	0.000	0.000	0.000	0.000	0.000
RIGHT WING	0.000	0.000	0.000	0.000	0.000	0.000
FREESTREAM	-422.908	0.000	-12118.064	-0.007	3040.153	-0.001
TOTAL	-422.908	0.000	-12118.064	-0.007	3040.153	-0.001
ENGINE PYLONS	-438.541	0.000	-9.652	0.000	1004.960	0.000
HORIZ. STAB.	-58.005	0.000	249.902	0.000	5232.124	0.000
VERT. STAB.						
FIN NO. 1	-24.266	-0.001	0.723	-0.002	87.499	-155.691
FIN NO. 2	-24.266	-0.001	0.723	-0.002	87.499	155.722
TOTAL	-48.532	-0.001	1.447	-0.005	174.997	0.031
LANDING GEAR						
MAIN AERO.	0.000	0.000	0.000	0.000	0.000	0.000
NOSE AERO.	0.000	0.000	0.000	0.000	0.000	0.000
DYNAMIC	0.000	0.000	0.000	0.000	0.000	0.000
TOTAL	0.000	0.000	0.000	0.000	0.000	0.000
JET THRUST						
LEFT ENG	-26.037	0.000	0.000	0.000	59.842	-418.768
RIGHT ENG	-26.037	0.000	0.000	0.000	59.842	418.768
TOTAL	-52.075	0.000	0.000	0.000	119.685	0.000
GROUND EFFECT						
FUSE-HOR STAB	0.000	0.000	0.000	0.000	0.000	0.000
TOTAL AIRFRAME	-1216.445	-0.001	-13015.515	-0.012	4336.775	0.030
ROTOR						
LEFT ROTOR	888.151	22.983	35.696	6468.565	-2494.138	14584.648
RIGHT ROTOR	888.151	-22.983	35.696	-6468.565	-2494.138	-14584.648
HUB SPINNER	-272.856	0.000	-6.019	0.000	653.363	0.000
TOTAL ROTOR	1503.446	0.000	65.374	0.000	-4334.914	0.000
TOTAL AIRCRAFT (BODY AXIS)	287.001	-0.001	-12950.142	-0.012	1.861	0.030
TOTAL AIRCRAFT (INERTIAL AXIS)	-0.799	-0.001	46.673	-0.012	1.861	0.029

APPENDIX L. MODIFIED JANRAD SCRIPT (MATLAB *.M) FILES

CONTENTS

```
%    STAB.M
%    NPS Helo Preliminary Design Program
%    Stability and Control Routines
%    Written by MAJ Walter M. Wirth, Jr.
%    September 1993
%

%    This program was designed as an interactive preliminary
%    design tool for stability and control analysis of a single
%    main rotor conventional or compound helicopter.  The
%    program provides stability derivatives, roots of the aircraft
%  ·  plant, plots of various control parameters as well as open loop
%    control bandwidths from control inputs to aircraft reponse.
%
%        *** Version 2.0 ***
%
%        Modification made by Capt Gary D. Klein, USMC
%
%        Modification made to incorporate the design and analysis of
%        a Tilt Rotor in Helo and Airplane modes (either/or but not
%        in transition/converson corridor)
%
clear
load temp
eval(['load ', filename1]);
clc
disp(' ')
disp(' ')
disp('             ***   STABILITY AND CONTROL ROUTINE   ***')
disp(' ')
disp(' ')
%pause(1)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%    *** If editing an existing file:  get file name, display edit
%        menu, allow changes to selected variables, and save under
%        desired file name.  Loads to and saves from current
%        directory as a .mat file. ***
  if exist('Ib');
    answer0=1;
  else
    answer0=2;
  end
  if answer0==1,
    check2=1;
      while check2 > 0
      clc
disp(' ')
disp('             ***   STABILITY AND CONTROL MENU   ***')
disp('             *** ADDITIONAL PARAMETERS (1 of 3) ***')
disp(' ')
disp(' Main Rotor')
disp(' 1. flapping mom of inertia  2. hub height above waterline')
disp(' 3. hub fuselage station     4. hub posn right of buttline')
disp(' 5. mast incidence           6. Flapping spring constant')
%
disp(' ')
disp(' Tail Rotor (enter zeros (0) if using NOTAR or Tilt Rotor)')
disp(' 7. height above waterline   8. hub fuselage station')
```

138

```
disp(' 9. posn right of buttline   10. number of blades')
disp('11. blade chord              12. blade radius')
disp('13. lift curve slope         14. rotational velocity')
disp('15. flap mom of inertia      16. delta-3 angle')
disp('17. blade twist')
%
disp(' ')
disp(' Vertical Fin')
disp('18. height above waterline  19. fuselage station')
disp('20. posn right of buttline  21. alpha zero lift')
disp('22. CL max                  23. dynamic pressure ratio')
disp('24. lift curve slope        25. Rudder effectiveness')
%
disp(' ')
disp('0. NO CHANGES')
      choice=input('Input the parameter to change: ');
      if choice==1,
        clc
        disp(' ')
        Ib
        temp1=Ib;
        Ib=input('Blade flapping moment of inertia (slug ft^2): ');
        if isempty(Ib),
            Ib=temp1;
        end
        clear temp1
      elseif choice==2,
        clc
        disp(' ')
        hmd
        temp1=hmd;
        hmd=input('Hub height above reference datum/waterline (ft): ');
        if isempty(hmd),
            hmd=temp1;
        end
        clear temp1
      elseif choice==3,
        clc
        disp(' ')
        lmd
        temp1=lmd;
        lmd=input('Hub fuselage station (ft): ');
        if isempty(lmd),
            lmd=temp1;
        end
        clear temp1
      elseif choice==4,
        clc
        disp(' ')
        ymd
        temp1=ymd;
        ymd=input('Hub position right of buttline (ft): ');
        if isempty(ymd),
            ymd=temp1;
        end
        clear temp1
      elseif choice==5,
        clc
        disp(' ')
        im*57.3
```

139

```
    temp1=im*57.3;
    im=input('Mast incidence (negative forward - deg): ')/57.3;
    if isempty(im),
        im=temp1/57.3;
    end
    clear temp1
 elseif choice==6,
   clc
   disp(' ')
   disp(['Kflpsprng/57.3 = ',num2str(Kflpsprng/57.3)])
   temp1=Kflpsprng;
   Kflpsprng=input('Hub flapping spring constant (ft-lbs/deg): ')*57.3;
   if isempty(Kflpsprng),
       Kflpsprng=temp1;
   end
   clear temp1
 elseif choice==7,
   clc
   disp(' ')
   htd
   temp1=htd;
   htd=input('Tail rotor height above reference datum/waterline (ft): ');
   if isempty(htd),
       htd=temp1;
   end
   clear temp1
 elseif choice==8,
   clc
   disp(' ')
   ltd
   temp1=ltd;
   ltd=input('Tail rotor fuselage station (ft): ');
   if isempty(ltd),
       ltd=temp1;
   end
   clear temp1
 elseif choice==9,
   clc
   disp(' ')
   ytd
   temp1=ytd;
   ytd=input('Tail rotor position right of buttline (ft): ');
   if isempty(ytd),
       ytd=temp1;
   end
   clear temp1
 elseif choice==10,
   clc
   disp(' ')
   bt
   temp1=bt;
   bt=input('Number of tail rotor blades: ');
   if isempty(bt),
       bt=temp1;
   end
   clear temp1
 elseif choice==11,
   clc
   disp(' ')
   cot
```

```
      temp1=cot;
      cot=input('Blade chord (ft): ');
      if isempty(cot),
         cot=temp1;
      end
      clear temp1
   elseif choice==12,
      clc
      disp(' ')
      Rt
      temp1=Rt;
      Rt=input('Tail rotor blade radius (ft): ');
      if isempty(Rt),
         Rt=temp1;
      end
      clear temp1
   elseif choice==13,
      clc
      disp(' ')
      at
      temp1=at;
      at=input('Average lift curve slope of tail rotor: ');
      if isempty(at),
         at=temp1;
      end
      clear temp1
   elseif choice==14,
      clc
      disp(' ')
      ohmt
      temp1=ohmt;
      ohmt=input('Rotational velocity of tail rotor (rad/sec): ');
      if isempty(ohmt),
         ohmt=temp1;
      end
      clear temp1
   elseif choice==15,
      clc
      disp(' ')
      Ibt
      temp1=Ibt;
      Ibt=input('Blade flapping moment of inertia (slug ft^2): ');
      if isempty(Ibt),
         Ibt=temp1;
      end
      clear temp1
   elseif choice==16,
      clc
      disp(' ')
      delta3*57.3
      temp1=delta3*57.3;
      delta3=input('Delta-3 angle (deg): ')/57.3;
      if isempty(delta3),
         delta3=temp1/57.3;
      end
      clear temp1
   elseif choice==17,
      clc
      disp(' ')
      theta1t*57.3
```

141

```
   temp1=theta1t*57.3;
   theta1t=input('Blade twist (deg): ')/57.3;
   if isempty(theta1t),
       theta1t=temp1/57.3;
   end
   clear temp1
elseif choice==18,
  clc
  disp(' ')
  hvd
  temp1=hvd;
  hvd=input('Height above reference datum/waterline (ft): ');
  if isempty(hvd),
      hvd=temp1;
  end
  clear temp1
elseif choice==19,
  clc
  disp(' ')
  lvd
  temp1=lvd;
  lvd=input('Fuselage station (ft): ');
  if isempty(lvd),
      lvd=temp1;
  end
  clear temp1
elseif choice==20,
  clc
  disp(' ')
  yvd
  temp1=yvd;
  yvd=input('Position right of buttline (ft): ');
  if isempty(yvd),
      yvd=temp1;
  end
  clear temp1
elseif choice==21,
  clc
  disp(' ')
  alplov*57.3
  temp1=alplov*57.3;
  alplov=input('Zero lift angle for vertical tail (deg): ')/57.3;
  if isempty(alplov),
      alplov=temp1/57.3;
  end
  clear temp1
elseif choice==22,
  clc
  disp(' ')
  clvertmax
  temp1=clvertmax;
  clvertmax=input('Maximum Cl for vertical tail: ');
  if isempty(clvertmax),
      clvertmax=temp1;
  end
  clear temp1
elseif choice==23,
  clc
  disp(' ')
  qvq
```

142

```
            temp1=qvq;
            qvq=input('Dynamic pressure ratio (pg 489 Prouty): ');
            if isempty(qvq),
                qvq=temp1;
            end
            clear temp1
        elseif choice==24,
            clc
            disp(' ')
            av
            temp1=av;
            av=input('Lift curve slope of vertical tail: ');
            if isempty(av),
                av=temp1;
            end
            clear temp1
        elseif choice==25,
            clc
            disp(' ')
            dclvddelr
            temp1=dclvddelr;
            dclvddelr=input('Change in (side force) lift wrt del r (1/rad): ');
            if isempty(dclvddelr),
                dclvddelr=temp1;
            end
            clear temp1
        elseif choice==0,
            check2=0;
            clc
        else
            disp(' ')
            disp('enter a displayed number ...press any key to continue')
            pause
        end        %if
    end                %while
%
    check2=1;
    while check2 > 0
        clc
disp(' ')
disp('            ***    STABILITY AND CONTROL MENU    ***')
disp('            *** ADDITIONAL PARAMETERS (2 of 3) ***')
%
disp(' ')
disp(' Horizontal Tail')
disp(' 1. height above waterline      2. fuselage station')
disp(' 3. posn right of buttline      4. alpha @ zero lift')
disp(' 5. angle of incidence          6. lift curve slope')
disp(' 7. dynamic pressure ratio      8. rotor downwash ratio')
disp(' 9. downwash wrt alpha ratio ')
%
disp(' ')
disp(' Wing')
disp('10. height above waterline     11. fuselage station')
disp('12. posn right of buttline     13. alpha @ zero lift')
disp('14. angle of incidence         15. lift curve slope')
disp('16. tip cord                   17. root cord')
disp('18. rotor downwash ratio       19. fuselage downwash ratio ')
disp('20. flaperon effectiveness')
%
```

143

```
disp(' ')
disp('CG location and Inertias/fuselage parameters')
disp('21. cg ht. above waterline     22. cg fuselage station')
disp('23. cg posn rt of buttline     24. Ixx')
disp('25. Iyy                        26. Izz')
disp('27. Ixz                        28. fuselage downwash ratio')
disp(' ')
%
disp('0. NO CHANGES')
      choice=input('Input the parameter to change: ');
      if choice==1
        clc
        disp(' ')
        hhd
        temp1=hhd;
        hhd=input('Height above reference datum/waterline (ft): ');
        if isempty(hhd),
            hhd=temp1;
        end
        clear temp1
      elseif choice==2,
        clc
        disp(' ')
        lhd
        temp1=lhd;
        lhd=input('Fuselage station (ft): ');
        if isempty(lhd),
            lhd=temp1;
        end
        clear temp1
      elseif choice==3,
        clc
        disp(' ')
        yhd
        temp1=yhd;
        yhd=input('Position right of buttline: ');
        if isempty(yhd),
            yhd=temp1;
        end
        clear temp1
      elseif choice==4,
        clc
        disp(' ')
        disp(['alploh = ',num2str(alploh*57.3)])
       disp(' ')
        temp1=alploh*57.3;
        alploh=input('Zero lift angle for horizontal tail (deg): ')/57.3;
        if isempty(alploh),
            alploh=temp1/57.3;
        end
        clear temp1
      elseif choice==5,
        clc
        disp(' ')
        disp(['ih = ',num2str(ih*57.3)])
       disp(' ')
        temp1=ih*57.3;
        ih=input('Angle of incidence of horizontal tail (deg): ')/57.3;
        if isempty(ih),
            ih=temp1/57.3;
```

144

```matlab
      end
      clear temp1
   elseif choice==6,
      clc
      disp(' ')
      ah
      temp1=ah;
      ah=input('Lift curve slope of horizontal tail: ');
      if isempty(ah),
         ah=temp1;
      end
      clear temp1
   elseif choice==7,
      clc
      disp(' ')
      qhq
      temp1=qhq;
      qhq=input('Dynamic pressure ratio (pg 489 Prouty): ');
      if isempty(qhq),
         qhq=temp1;
      end
      clear temp1
   elseif choice==8,
      clc
      disp(' ')
      vhv1
      temp1=vhv1;
      vhv1=input('Rotor downwash ratio (pg 489 Prouty): ');
      if isempty(vhv1),
         vhv1=temp1;
      end
      clear temp1
   elseif choice==9,
      clc
      disp(' ')
      depsdalph
      temp1=depsdalph;
      depsdalph=input('Fuselage downwash ratio (pg 489 Prouty): ');
      if isempty(depsdalph),
         depsdalph=temp1;
      end
      clear temp1
   elseif choice==10
      clc
      disp(' ')
      hwd
      temp1=hwd;
      hwd=input('Height above reference datum/waterline (ft): ');
      if isempty(hwd),
         hwd=temp1;
      end
      clear temp1
   elseif choice==11
      clc
      disp(' ')
      lwd
      temp1=lwd;
      lwd=input('Fuselage station (ft): ');
      if isempty(lwd),
         lwd=temp1;
```

145

```
      end
      clear temp1
    elseif choice==12
      clc
      disp(' ')
      ywd
      temp1=ywd;
      ywd=input('Position right of buttline (ft): ');
      if isempty(ywd),
          ywd=temp1;
      end
      clear temp1
    elseif choice==13,
      clc
      disp(' ')
      disp(['alplow = ',num2str(alplow*57.3)])
   disp(' ')
      temp1=alplow*57.3;
      alplow=input('Zero lift angle for wing (deg): ')/57.3;
      if isempty(alplow),
          alplow=temp1/57.3;
      end
      clear temp1
    elseif choice==14,
      clc
      disp(' ')
      iw*57.3
      temp1=iw*57.3;
      iw=input('Angle of incidence of wing (deg): ')/57.3;
      if isempty(iw),
          iw=temp1/57.3;
      end
      clear temp1
    elseif choice==15,
      clc
      disp(' ')
      aw
      temp1=aw;
      aw=input('Lift curve slope of wing: ');
      if isempty(aw),
          aw=temp1;
      end
      clear temp1
    elseif choice==16,
      clc
      disp(' ')
      ctw
      temp1=ctw;
      ctw=input('Tip cord (ft): ');
      if isempty(ctw),
          ctw=temp1;
      end
      clear temp1
    elseif choice==17,
      clc
      disp(' ')
      crw
      temp1=crw;
      crw=input('Root cord (ft): ');
      if isempty(crw),
```

146

```
        crw=temp1;
      end
      clear temp1
    elseif choice==18,
      clc
      disp(' ')
      vwv1
      temp1=vwv1;
      vwv1=input('Rotor downwash ratio (pg 489 Prouty): ');
      if isempty(vwv1),
         vwv1=temp1;
      end
      clear temp1
    elseif choice==19,
      clc
      disp(' ')
      detafdalpfw
      temp1=detafdalpfw;
      detafdalpfw=input('Fuselage downwash ratio (pg 489 Prouty): ');
      if isempty(detafdalpfw),
         detafdalpfw=temp1;
      end
      clear temp1
    elseif choice==20,
      clc
      disp(' ')
      dclwddelf
      temp1=dclwddelf;
      dclwddelf=input('Change in roll moment coeff. wrt flaperon defl.
(1/rad): ');
      if isempty(dclwddelf),
         dclwddelf=temp1;
      end
      clear temp1
    elseif choice==21,
      clc
      disp(' ')
      zcg
      temp1=zcg;
      zcg=input('CG height above reference datum/waterline (ft): ');
      if isempty(zcg),
         zcg=temp1;
      end
      clear temp1
    elseif choice==22,
      clc
      disp(' ')
      xcg
      temp1=xcg;
      xcg=input('CG Fuselage station (ft): ');
      if isempty(xcg),
         xcg=temp1;
      end
      clear temp1
    elseif choice==23,
      clc
      disp(' ')
      ycg
      temp1=ycg;
      ycg=input('CG position right of buttline (ft): ');
```

```
    if isempty(ycg),
        ycg=temp1;
    end
    clear temp1
elseif choice==24,
    clc
    disp(' ')
    Ixx
    temp1=Ixx;
    Ixx=input('Ixx (slug ft^2): ');
    if isempty(Ixx),
        Ixx=temp1;
    end
    clear temp1
elseif choice==25,
    clc
    disp(' ')
    Iyy
    temp1=Iyy;
    Iyy=input('Iyy (slug ft^2): ');
    if isempty(Iyy),
        Iyy=temp1;
    end
    clear temp1
elseif choice==26,
    clc
    disp(' ')
    Izz
    temp1=Izz;
    Izz=input('Izz (slug ft^2): ');
    if isempty(Izz),
        Izz=temp1;
    end
    clear temp1
elseif choice==27,
    clc
    disp(' ')
    Ixz
    temp1=Ixz;
    Ixz=input('Ixz (slug ft^2): ');
    if isempty(Ixz),
        Ixz=temp1;
    end
    clear temp1
elseif choice==28,
    clc
    disp(' ')
    vfv1
    temp1=vfv1;
    vfv1=input('Downwash ratio for fuselage (page 513 Prouty): ');
    if isempty(vfv1),
        vfv1=temp1;
    end
    clear temp1
elseif choice==0,
    check2=0;
    clc
else
    disp(' ')
    disp('enter a displayed number ... press any key to continue')
```

148

```
        pause
      end     %if
    end              %while
%
    check2=1;
    while check2 > 0
      clc
disp(' ')
disp('          ***   STABILITY AND CONTROL MENU   ***')
disp('          *** ADDITIONAL PARAMETERS (3 of 3) ***')
%
disp(' ')
disp(' NOTAR if available (enter zeros if using tail or tilt rotor)')
disp(' 1. height above waterline    2. boom fuselage station')
disp(' 3. boom position left ref    4. NOTAR diameter ')
disp(' 5. swirl angle at boom       6. NOTAR max force')
disp(' 7. thruster fuselage station')
%
disp(' ')
disp(' Tilt Rotor (enter zeros if using tail rotor or NOTAR)')
disp(' 8. Fuselage CP location       9. Fuselage angle @ zero lift ')
disp('10. Fuselage lift slope       11. Fuselage Cmo')
disp('12. Fuselage moment slope     13. Wing aero. center')
disp('14. Wing sweep                15. Wing dihedral')
disp('16. Wing moment coefficient   17. Downwash angle @ zero alpha')
disp('18. Horiz. stab. span eff.    19. Elevator effectiveness ')
%
disp(' ')
disp(' Rigging')
disp('20. B1 main/in defl (del e)   21. A1 main/in defl (del a)')
disp('22. theta0m/in defl (del c)   23. theta0t/pedal defl (del r or p)')
disp('24. NOTAR sleeve twist/defl   25. max rudder defl')
disp('26. Aileron/in defl (del a)   27. Elevator/in defl (del e)')
disp('28. Rudder/in defl (del r)')
disp(' ')
disp('0. NO CHANGES')
      choice=input('Input the parameter to change: ');
      if choice==1,
        clc
        disp(' ')
        htnd
        temp1=htnd;
        htnd=input('Height above reference datum/waterline (ft): ');
        if isempty(htnd),
           htnd=temp1;
        end
        clear temp1
      elseif choice==2,
        clc
        disp(' ')
        ltnd
        temp1=ltnd;
        ltnd=input('Fuselage station (ft): ');
        if isempty(ltnd),
           ltnd=temp1;
        end
        clear temp1
      elseif choice==3,
        clc
        disp(' ')
```

149

```
      ytnd
      temp1=ytnd;
      ytnd=input('Position right of buttline (ft): ');
      if isempty(ytnd),
          ytnd=temp1;
      end
      clear temp1
    elseif choice==4,
      clc
      disp(' ')
      dian
      temp1=dian;
      dian=input('NOTAR boom diameter (ft): ');
      if isempty(dian),
          dian=temp1;
      end
      clear temp1
    elseif choice==5,
      clc
      disp(' ')
      swirl*57.3
      temp1=swirl*57.3;
      swirl=input('Swirl angle at boom(deg): ')/57.3;
      if isempty(swirl),
          swirl=temp1/57.3;
      end
      clear temp1
    elseif choice==6,
      clc
      disp(' ')
      Ytmaxn
      temp1=Ytmaxn;
      Ytmaxn=input('Maximum NOTAR thruster force (lbs): ');
      if isempty(Ytmaxn),
          Ytmaxn=temp1;
      end
      clear temp1
    elseif choice==7,
      clc
      disp(' ')
      lttnd
      temp1=lttnd;
      lttnd=input('Thruster fuselage station (ft): ');
      if isempty(lttnd),
          lttnd=temp1;
      end
      clear temp1
    elseif choice==8
      clc
      disp(' ')
      lfd
      temp1=lfd;
      lfd=input('Fuselage station of (fuselage) Center of Pressure (ft): ');
      if isempty(lfd),
          lfd=temp1;
      end
      clear temp1
    elseif choice==9
      clc
      disp(' ')
```

150

```matlab
      disp(['alplof = ',num2str(alplof*57.3),' deg'])
      temp1=alplof;
      alplof=input('Fuselage angle @ zero lift (degrees) : ')/57.3;
      if isempty(alplof),
          alplof=temp1;
      end
      clear temp1
   elseif choice==10,
      clc
      disp(' ')
      af
      temp1=af;
      af=input('Lift curve slope of fuselage (1/rad): ');
      if isempty(af),
          af=temp1;
      end
      clear temp1
   elseif choice==11
      clc
      disp(' ')
      cmof
      temp1=cmof;
      cmof=input('Fus. mom. coef. @ zero alpha (ref. to Aw, cw): ');
      if isempty(cmof),
          cmof=temp1;
      end
      clear temp1
   elseif choice==12,
      clc
      disp(' ')
      cmalpf
      temp1=cmalpf;
      cmalpf=input('Slope of fus. moment coef. wrt alpha curve (1/rad): ');
      if isempty(cmalpf),
          cmalpf=temp1;
      end
      clear temp1
   elseif choice==13,
      clc
      disp(' ')
      acw
      temp1=acw;
      acw=input('Wing Aerodynamic Center location (% cw): ');
      if isempty(acw),
          acw=temp1;
      end
      clear temp1
   elseif choice==14,
      clc
      disp(' ')
      disp(['lambda = ',num2str(lambda*180/pi)])
      temp1=lambda;
      lambda=input('Wing sweep angle (deg): ');
      if isempty(lambda),
          lambda=temp1;
      end
      clear temp1
   elseif choice==15,
      clc
      disp(' ')
```

151

```
    disp(['dih = ',num2str(dih*180/pi)])
    temp1=dih;
    dih=input('Wing Dihedral angle (deg): ');
    if isempty(dih),
       dih=temp1;
    end
    clear temp1
 elseif choice==16,
    clc
    disp(' ')
    cmow
    temp1=cmow;
    cmow=input('Wing Moment Coeff @ zero lift: ');
    if isempty(cmow),
       cmow=temp1;
    end
    clear temp1
 elseif choice==17,
    clc
    disp(' ')
    epso
    temp1=epso;
    epso=input('Downwash angle @ zero alpha (rad): ');
    if isempty(epso),
       epso=temp1;
    end
    clear temp1
 elseif choice==18
    clc
    disp(' ')
 disp('delih = 1/e - 1')
    delih
    temp1=delih;
    delih=1/(input('Horizontal Tail Span Efficiency (e): '))-1;
    if isempty(delih),
       delih=temp1;
    end
    clear temp1
 elseif choice==19,
    clc
    disp(' ')
    dclhddeleh
    temp1=dclhddeleh;
    dclhddeleh=input('Change in H-stab Cl wrt elevator angle (1/rad): ');
    if isempty(dclhddeleh),
       dclhddeleh=temp1;
    end
    clear temp1
 elseif choice==20,
    clc
    disp(['db1mddele = ',num2str(db1mddele*57.3)])
    temp1=db1mddele*57.3;
    db1mddele=input('Long cyclic pitch per inch defl (deg/in): ')/57.3;
    if isempty(db1mddele),
       db1mddele=temp1/57.3;
    end
    clear temp1
 elseif choice==21,
    clc
    disp(['da1mddela = ',num2str(da1mddela*57.3)])
```

152

```
      temp1=da1mddela*57.3;
      da1mddela=input('Lateral cyclic pitch per inch defl (deg/in): ')/57.3;
      if isempty(da1mddela),
         da1mddela=temp1/57.3;
      end
      clear temp1
   elseif choice==22,
      clc
      disp(['dthetomddelc = ',num2str(dthetomddelc*57.3)])
      temp1=dthetomddelc*57.3;
      dthetomddelc=input('Collective pitch per inch defl (deg/in): ')/57.3;
      if isempty(dthetomddelc),
         dthetomddelc=temp1/57.3;
      end
      clear temp1
   elseif choice==23,
      clc
      disp(['dthetotddelp = ',num2str(dthetotddelp*57.3)])
      temp1=dthetotddelp*57.3;
      disp('Tail rotor pitch change per inch defl or percentage of twist')
      disp('Enter 0 (zero) if using NOTAR')
      dthetotddelp=input('(deg/in or deg/deg of twist): ')/57.3;
      if isempty(dthetotddelp),
         dthetotddelp=temp1/57.3;
      end
      clear temp1
   elseif choice==24,
      clc
      disp(['sidearm/2 = ',num2str(sidearm/2)])
      temp1=sidearm/2;
      disp('Maximum deflection of anti-torque from neutral for NOTAR, enter')
      sidearm=input('1000 if using tail rotor (deg or inch travel): ')*2;
      if isempty(sidearm),
         sidearm=temp1*2;
      if sidearm==0,sidearm=1e3,end
      end
      clear temp1
      dphinddelp=pi/sidearm; %% pi rad sleeve twist/sidearm defl
   elseif choice==25,
      clc
      disp(' ')
      maxr
      temp1=maxr;
      disp('Displacement of anti-torque control until full rudder')
      disp('  deflection.  Enter 0 (zero) if rudder is fixed')
      maxr=input('   (deg or inch travel): ');
      if isempty(maxr),
         maxr=temp1;
      end
      clear temp1
   elseif choice==26,
      clc
      disp(' ')
      disp(['ddeladlat = ',num2str(ddeladlat*57.3),' deg/in'])
      temp1=ddeladlat*57.3;
      ddeladlat=input('Aileron angle per inch defl (deg/in): ')/57.3;
      if isempty(ddeladlat),
         ddeladlat=temp1/57.3;
      end
      clear temp1
```

153

```matlab
        elseif choice==27,
          clc
          disp(' ')
          disp(['ddeledlong = ',num2str(ddeledlong*57.3),' deg/in'])
          temp1=ddeledlong*57.3;
          ddeledlong=input('Elevator angle per inch defl (deg/in): ')/57.3;
          if isempty(ddeledlong),
              ddeledlong=temp1/57.3;
          end
          clear temp1
        elseif choice==28,
          clc
          disp(['ddelrddelp = ',num2str(ddelrddelp*57.3),' deg/in'])
          temp1=ddelrddelp*57.3;
          disp(' ')
          disp('Enter 0 (zero) if using NOTAR or tailrotor')
          ddelrddelp=input('Rudder angle per inch defl of pedals: ')/57.3;
          if isempty(ddelrddelp),
              ddelrddelp=temp1/57.3;
          end
          clear temp1
        elseif choice==0,
          clc
          check2=0;
        else
          disp(' ')
          disp('enter a displayed number ... press any key to continue')
          pause
        end
      end
%
disp(' ')
disp(' ')
disp('                   *** SAVE INSTRUCTIONS ***')
disp(' ')
disp('A. Save the new data to a specified file name.')
disp('B. Do not use an extension or quotations.')
disp('C. Use letter/number combinations of 6 characters or less.')
disp('D. The file will be saved with a ".mat" extension.')
disp(' ')
disp('E. If you made no changes or want the same name, press enter.')
disp (' ')
disp('ex: desig2')
     filename=filename1;
     filename1=input('save file as: ','s');
     if isempty(filename1),
        filename1=filename;
     end
     clear check
     eval(['save ',filename1]);
     check=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%    *** If creating a new file:  get input for required variables
%        and save under desired file name.  Saves to current
%        directory as a .mat file. ***

 else
  check4=1;
```

154

```
while check4>0;
clc
disp('Do you want to design a Tail Rotor, Tilt Rotor or NOTAR?')
temp=input('Tail Rotor = 0, NOTAR = 1, Tilt Rotor = 2: ');
  if temp==0
    tiltr=0;
    notar=0;
    check4=0;
  elseif temp==1
    notar=1;
    tiltr=0;
    check4=0;
  elseif temp==2
    tiltr=1;
    notar=0;
    check4=0;
  else
    disp(' ')
    disp('Enter a 0, 1 or 2')
    disp('press any key to continue...')
    pause
  end                %if
end          %while
check4=1;
while check4>0;
clc
disp('Do you want to use a controlable vertical tail?')
temp=input('No=0, Yes=1: ');
  if temp==0
    ctail=0;
    check4=0;
  elseif temp==1
    ctail=1;
    check4=0;
  else
    disp(' ')
    disp('Enter a 0 or 1')
    disp('press any key to continue...')
    pause
  end                %if
end          %while
if Swing<.1
  wing=0;
else
  wing=1;
end
  clc
  disp('Main rotor')
  disp(' ')
      Ib=input('Blade flapping moment of inertia (slug ft^2): ');
      hmd=input('Hub height above reference datum/waterline (ft): ');
      lmd=input('Hub fuselage station (ft): ');
      ymd=input('Hub position right of buttline (ft): ');
      im=input('Mast incidence (negative forward - deg): ')/57.3;
    Kflpsprng=input('Hub flapping spring constant (ft-lbs/deg): ')*57.3;
  clc
if notar==0 & tiltr==0
  disp('Tail rotor')
  disp(' ')
      htd=input('Tail rotor height above reference datum/waterline (ft): ');
```

155

```
            ltd=input('Tail rotor fuselage station (ft): ');
            ytd=input('Tail rotor position right of buttline (ft): ');
            bt=input('Number of tail rotor blades: ');
            cot=input('Tail rotor blade chord (ft): ');
            Rt=input('Tail rotor blade radius (ft): ');
            at=input('Average lift curve slope of tail rotor: ');
            ohmt=input('Rotational velocity of tail rotor (rad/sec): ');
            Ibt=input('Tail rotor blade flapping moment of inertia (slug ft^2): ');
            delta3=input('Delta-3 angle (deg): ')/57.3;
            theta1t=input('Blade twist (deg): ')/57.3;
            htnd=0;ltnd=0;ytnd=0;dian=0;swirl=0;Ytmaxn=0;lttnd=0;
    elseif notar==1
      clc
      disp('NOTAR')
      disp(' ')
            htnd=input('Height above reference datum/waterline (ft): ');
            ltnd=input('Fuselage station (ft): ');
            ytnd=input('Position right of buttline (ft): ');            .
            dian=input('NOTAR boom diameter (ft): ');
            swirl=input('Swirl angle at boom (deg): ')/57.3;
            Ytmaxn=input('Maximum NOTAR thruster force (lbs): ');
            lttnd=input('Thruster fuselage stationerence (ft): ');
%
            htd=0;ltd=0;ytd=0;bt=0;cot=0;Rt=0;at=0;ohmt=0;
            Ibt=0;delta3=0;theta1t=0;
          ddeladlat=0;ddeledlong=0;ddelrddelp=0;
%
    elseif tiltr==1
      clc
      disp('Tilt Rotor')
      disp(' ')
            lfd=input('Fuselage station of (fuselage) Center of Pressure (ft): ');
            alplof=input('Fuselage angle @ zero lift (degrees) : ')/57.3;
            af=input('Lift curve slope of fuselage (1/rad): ');
            cmof=input('Fus. mom. coef. @ zero alpha (ref. to Aw & cw): ');
            cmalpf=input('Slope of fus. moment coef. wrt alpha curve (1/rad): ');
          delih=1/(input('Horizontal Tail Span Efficiency (e): '))-1;
            epso=input('Downwash angle @ zero alpha (rad): ');
            dclwddelf=input('Change in roll moment coeff. wrt flaperon defl.
(1/rad): ');
            dclhddeleh=input('Change in H-stab Cl wrt elevator angle (1/rad): ');
            acw=input('Wing Aerodynamic Center location (% cw): ');
            lambda=input('Wing sweep angle (deg): ')*pi/180;
            dih=input('Wing Dihedral angle (deg): ')*pi/180;
            cmow=input('Wing Moment Coeff @ zero lift: ');
            ddeladlat=input('Aileron angle per inch defl (deg/in): ')/57.3;
            dthetomddelc=input('Collective thrust per inch defl (deg/in): ')/57.3;
            ddeledlong=input('Elevator angle per inch defl (deg/in): ')/57.3;
            ddelrddelp=input('Rudder angle per inch defl of pedals: ')/57.3;
%
            htd=0;ltd=0;ytd=0;bt=0;cot=0;Rt=0;at=0;ohmt=0;
            Ibt=0;delta3=0;theta1t=0;
          htnd=0;ltnd=0;ytnd=0;dian=0;swirl=0;Ytmaxn=0;lttnd=0;
    end         %if
      clc
      disp('Vertical tail')
      disp(' ')
            hvd=input('Height above reference datum/waterline (ft): ');
            lvd=input('Fuselage station (ft): ');
            yvd=input('Position right of buttline (ft): ');
```

```
        alplov=input('Zero lift angle for vertical tail (deg): ')/57.3;
        clvertmax=input('Maximum Cl for vertical tail: ');
        qvq=input('Dynamic pressure ratio for tail (pg 489 Prouty): ');
        av=input('Lift curve slope of vertical tail: ');
      crv=input('Vert. tail root chord (ft): ');
      ctv=input('Vert. tail tip chord (ft): ');
      delih=1/(input('Vert. tail span efficiency factor (e): '))-1;
      cfcv=input('Rudder chord length (% cv): ');
  clc
  disp('Horizontal tail')
  disp(' ')
      hhd=input('Height above reference datum/waterline (ft): ');
      lhd=input('Fuselage station (ft): ');
      yhd=input('Position right of buttline: ');
      alploh=input('Zero lift angle for horizontal tail (deg): ')/57.3;
      ih=input('Angle of incidence of horizontal tail (deg): ')/57.3;
      ah=input('Lift curve slope of horizontal tail: ');
      qhq=input('Dynamic pressure ratio for tail (pg 489 Prouty): ');
      vhv1=input('Rotor downwash ratio for h-tail(pg 489 Prouty): ');
      depsdalph=input('Fuselage downwash ratio for h-tail (pg 489 Prouty): ');
if wing==1
  clc
  disp('Wing')
  disp(' ')
      hwd=input('Height above reference datum/waterline (ft): ');
      lwd=input('Fuselage station (ft): ');
      ywd=input('Position right of buttline (ft): ');
      alplow=input('Zero lift angle for wing (deg): ')/57.3;
      iw=input('Angle of incidence of wing (deg): ')/57.3;
      aw=input('Lift curve slope of wing: ');
      ctw=input('Tip cord (ft): ');
      crw=input('Root cord (ft): ');
      vwv1=input('Rotor downwash ratio for wing (pg 489 Prouty): ');
      detafdalpfw=input('Fuselage downwash ratio for wing (pg 489 Prouty): ');
elseif wing==0
      hwd=0;lwd=0;ywd=0;alplow=0;iw=0;aw=0;ctw=0;crw=0;
      vwv1=0;detafdalpfw=0;
end          %if
  clc
  disp('CG location')
  disp(' ')
      zcg=input('CG height above reference datum/waterline (ft): ');
      xcg=input('CG Fuselage station (ft): ');
      ycg=input('CG position right of buttline (ft): ');
  clc
  disp('Fuselage moments of inertia/downwash parameter')
  disp(' ')
      Ixx=input('Ixx (slug ft^2): ');
      Iyy=input('Iyy (slug ft^2): ');
      Izz=input('Izz (slug ft^2): ');
      Ixz=input('Ixz (slug ft^2): ');
      vfv1=input('Downwash ratio for fuselage (page 513 Prouty): ');
  clc
  disp(' Rigging')
  disp(' ')
      db1mddele=input('Long cyclic pitch per inch defl (deg/in): ')/57.3;
      da1mddela=input('Lateral cyclic pitch per inch defl (deg/in): ')/57.3;
      dthetomddelc=input('Collective pitch per inch defl (deg/in): ')/57.3;
  if notar==0
      disp(' ')
```

157

```
            disp('Tail rotor pitch change per inch defl or percentage of twist')
            dthetotddelp=input('   (deg/in or deg/deg of twist): ')/57.3;
            dphinddelp=0;sidearm=1000;
        elseif notar==1
            disp(' ')
            disp('Max deflection of anti-torque from neutral for NOTAR, enter ')
            sidearm=input('  1000 if using tail rotor (deg or inch travel):  ')*2;
            if sidearm==0,sidearm=1000,end
            dphinddelp=pi/sidearm;
        end
        if ctail==1
            disp(' ')
            disp('Displacement of anti-torque control until full rudder')
            maxr=input('   deflection (deg or inch travel): ');
          dclvddelr=input('Change in (side) lift wrt del r (1/rad): ');
        elseif ctail==0
            maxr=0;
            dclvddelr=0;
        end
        clc
        eval(['save ',filename1]);
        save stabtemp filename1
        check=0;
    end
    if dian==0 & tiltr==0;
      notar=0;
      if bt==0;
        disp('  You must have a tail rotor or NOTAR/thruster! ')
        check=1;
      end
    else
      if tiltr~=1
        notar=1;
      end
    end
end             %if answer0==?
clc
disp(' ')
disp(' ')
disp('                          *** DATA ENTRY COMPLETE ***')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' ')
disp('              *** EVALUATING STABILITY DERIVATIVES ***')
%pause(1)
if Vinf<20
   if tiltr==1
     tltrhovr
   else
     hover          % call hover routine
   end
elseif Vinf>=20
   if tiltr==1
     tltrcrus
   else             % call cruise routine
     cruise
   end
end
stabout
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%    STABOUT.M
%
%    Called By: STAB.M
%
%    calls
%    CMDBWPLH
%    CMDBWPLC
%    TMRESPH
%    TMRESPC
%
%    *** Stability and Control Output Subroutine ***
%
%        Modified 1996 to incorporate tiltrotor parameters
%        and Time response plots
%        By:  Capt Gary Klein, USMC
%
pack
format compact
clc
disp(' ')
disp(' ')
disp('Do you want the results displayed on screen?')
disp(' ')
disp('  NOTE: if you want a hard copy of the plots, you must')
disp('  select (1) and view them on the screen first.')
flag=1;
answer=input('1. yes   2. no   >> ');
while flag>0
  if answer == 2,
    flag=0;
  elseif answer == 1,
%
% output to screen
%
  clc
  disp(' ')
  disp('           *** STABILITY AND CONTROL PROGRAM ***')
  disp('                   *** SCREEN VIEW MENU ***')
  disp(' ')
  disp('What do you want to see?')
  disp(' ')
  disp('1. Input data.')
  disp('2. Calculated data.')
  disp('3. State Matrices.')
  disp('4. Eigenvalues of the plants and plots of the roots.')
  disp('5. Key control parameters.')
  disp('6. Open loop transfer (Bode-Magnitude) plots.')
  disp('7. Open loop Time Response to Control inputs')
  disp(' ')
  disp('0. Exit screen view.')
  disp(' ')
    choice=input('Enter a number; ');
    if choice==1,
clc
disp(' ')
disp('              *** INPUT DATA (screen 1 of 8) ***')
eval(['disp('                                ',filename1,''')'])
disp(' ')
disp('                        Flight Conditions')
disp(' ')
```

159

```
fprintf('                    Forward velocity = %6.0f kts\n',Vinf/1.69)
fprintf('                        Temperature = %6.0f degs F\n',temp)
fprintf('                  Pressure altitude = %6.0f ft\n',PA)
fprintf('                   Auxiliary thrust = %6.0f lbs\n',Taux)
disp(' ')
disp('                            Fuselage')
disp(' ')
fprintf('                       Gross weight = %6.0f lbs\n',GW)
fprintf('      Equivalent flat plate area = %6.1f ft^2\n',Afh)
fprintf('          Vertical projected area = %6.1f ft^2\n',Afv)
fprintf('      Center of Pressure station = %6.1f ft\n',lfd)
fprintf('      Fuselage alpha @ zero lift = %6.1f degrees\n',alplof*57.3)
fprintf(' Lift curve slope of fuselage = %6.3f 1/rad\n',af)
fprintf(' Moment coefficient @ 0 alpha = %6.3f (referenced to Aw, cw)\n',cmof)
fprintf('     Moment coeff./alpha slope = %6.3f 1/rad\n',cmalpf)
fprintf('      CG height above waterline = %6.1f ft\n',zcg)
fprintf(' Aircraft CG fuselage station = %6.1f ft\n',xcg)
fprintf('CG position right of buttline = %6.1f ft\n',ycg)
fprintf('                                Ixx = %6.1f slug ft^2\n',Ixx)
fprintf('                                Iyy = %6.1f slug ft^2\n',Iyy)
fprintf('                                Izz = %6.1f slug ft^2\n',Izz)
fprintf('                                Ixz = %6.1f slug ft^2\n',Ixz)
fprintf('                   Downwash ratio = %6.2f \n',vfv1)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('       *** INPUT DATA CONTINUED (screen 2 of 8) ***')
eval(['disp(''                         ',filename1,''')'])
disp(' ')
disp('                         Main Rotor')
disp(' ')
fprintf('             Number of blades = %6.0f \n',b)
fprintf('                 Rotor radius = %6.1f ft\n',R)
if tiltr==0
  fprintf('            Ave blade chord = %6.1f
ft\n',(sum(cblade)/length(cblade)));
end
fprintf('                  Blade twist = %6.2f degs\n',twist*57.3)
if Airfoil==1,
   disp('             Blade airfoil =  HH-02')
elseif Airfoil==2,
   disp('             Blade airfoil =  VR-12')
else,
   disp('             Blade airfoil =  NACA 0012')
end
fprintf('       Blade lift curve slope = %6.2f \n',a)
fprintf('                 Blade weight = %6.1f lbs\n',wblade)
fprintf('          Rotational velocity = %6.2f rpm\n',omega*30/pi)
fprintf('            Blade grip length = %6.1f ft\n',grip)
fprintf('                  Hinge offset = %6.1f ft\n',e)
fprintf('     Flapping spring constant = %6.1f ft-lbs/deg\n',Kflpsprng/57.3)
fprintf('    Flapping moment of inertia = %6.1f slug ft^2\n',Ib)
fprintf('    Hub height above waterline = %6.1f ft\n',hmd)
fprintf('          Hub fuselage station = %6.1f ft\n',lmd)
fprintf('  Hub position rt of buttline = %6.1f ft\n',ymd)
fprintf('                Mast incidence = %6.2f deg\n',im*57.3)
disp(' ')
disp(' ')
```

160

```
disp('press any key to continue...')
pause
clc
disp(' ')
disp('            *** INPUT DATA CONTINUED (screen 3 of 8) ***')
eval(['disp(''                        ',filename1,''')'])
disp(' ')
disp('              Tail rotor (zeros if using NOTAR)')
disp(' ')
fprintf('            Number of blades = %6.1f \n',bt)
fprintf('                 Blade chord = %6.1f ft\n',cot)
fprintf('                Blade radius = %6.1f ft\n',Rt)
fprintf('            Lift curve slope = %6.2f \n',at)
fprintf('         Rotational velocity = %6.2f rad/sec\n',ohmt)
fprintf('    Flapping moment of inertia = %6.1f slug ft^2\n',Ibt)
fprintf('               Delta-3 angle = %6.2f deg\n',delta3*57.3)
fprintf('                  Blade twist = %6.2f deg\n',theta1t*57.3)
fprintf('  Hub height above waterline = %6.1f ft\n',htd)
fprintf('         Hub fuselage station = %6.1f ft\n',ltd)
fprintf('  Hub position rt of buttline = %6.1f ft\n',ytd)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('            *** INPUT DATA CONTINUED (screen 4 of 8) ***')
eval(['disp(''                        ',filename1,''')'])
disp(' ')
disp('              NOTAR (zeros if using tail or tilt rotor)')
disp(' ')
fprintf('        Height above waterline = %6.1f ft^2\n',htnd)
fprintf('             Fuselage station = %6.1f ft^2\n',ltnd)
fprintf('    Position right of buttline = %6.1f ft^2\n',ytnd)
fprintf('         NOTAR boom diameter = %6.1f ft^2\n',dian)
fprintf('         Swirl angle at boom = %6.2f deg\n',swirl*57.3)
fprintf('       Maximum thruster force = %6.1f lbs\n',Ytmaxn)
fprintf('       Thrust fuselage station = %6.1f ft^2\n',lttnd)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('            *** INPUT DATA CONTINUED (screen 5 of 8) ***')
eval(['disp(''                        ',filename1,''')'])
disp(' ')
disp('                         Wing')
disp(' ')
fprintf('                        Area = %6.1f ft^2\n',Swing)
fprintf('                        Span = %6.1f ft\n',bwing)
fprintf('                          CL = %6.2f \n',CLwing)
fprintf('                         CDo = %6.4f \n',CDowing)
fprintf('                     Tip cord = %6.1f ft\n',ctw)
fprintf('                    Root cord = %6.1f ft\n',crw)
fprintf('       Wing efficiency factor = %6.2f \n',ewing)
fprintf('             Zero lift angle = %6.2f deg\n',alplow*57.3)
fprintf('           Angle of incidence = %6.2f deg\n',iw*57.3)
fprintf('            Wing sweep angle = %6.2f deg\n',lambda*57.3)
fprintf('          Wing Dihedral angle = %6.2f deg\n',dih*57.3)
fprintf('              Lift curve slope = %6.2f per radian\n',aw)
fprintf(' Delta Clw per flaperon angle = %6.2f per radian\n',dclwddelf)
```

161

```
fprintf('Wing Moment Coeff @ zero lift = %6.2f \n',cmow)
fprintf('        Rotor downwash ratio = %6.2f \n',vwv1)
fprintf('     Fuselage downwash ratio = %6.2f \n',detafdalpfw)
fprintf('  Downwash angle @ zero alpha = %6.2f radians\n',epso)
fprintf('        Height above waterline = %6.1f ft\n',hwd)
fprintf('Fuselage station (of wing CP) = %6.1f ft\n',lwd)
fprintf('CP Position right of buttline = %6.1f ft\n',ywd)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('        *** INPUT DATA CONTINUED (screen 6 of 8) ***')
eval(['disp(''                        ',filename1,''')'])
disp(' ')
disp('                      Horizontal tail')
disp(' ')
fprintf('                          Area = %6.1f ft^2\n',Shoriz)
fprintf('                          Span = %6.1f ft\n',bhoriz)
fprintf('                            CL = %6.2f \n',CLhoriz)
fprintf('                           CDo = %6.4f \n',CDohoriz)
fprintf('               Zero lift angle = %6.2f deg\n',alploh*57.3)
fprintf('           Angle of incidence = %6.2f deg\n',iw*57.3)
fprintf('             Lift curve slope = %6.1f \n',ah)
fprintf('        Height above waterline = %6.1f ft\n',hhd)
fprintf('              Fuselage station = %6.1f ft\n',lhd)
fprintf('   Position right of buttline = %6.1f ft\n',yhd)
fprintf('        Dynamic pressure ratio = %6.2f \n',qhq)
fprintf('          Rotor downwash ratio = %6.2f \n',vhv1)
fprintf('     Fuselage downwash ratio = %6.2f \n',depsdalph)
fprintf('   H-Tail Span Efficiency (e) = %6.2f \n',1/(1+delih))
fprintf('Delta Clh per elevator angle  = %6.2f per radian\n',dclhddeleh)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('        *** INPUT DATA CONTINUED (screen 7 of 8) ***')
eval(['disp(''                        ',filename1,''')'])
disp(' ')
disp('                      Vertical tail')
disp(' ')
fprintf('                          Area = %6.1f ft^2\n',Svert)
fprintf('                          Span = %6.1f ft\n',bvert)
fprintf('                            CL = %6.2f \n',CLvert)
fprintf('                           CDo = %6.4f \n',CDovert)
fprintf('        Height above waterline = %6.1f ft\n',hvd)
fprintf('              Fuselage station = %6.1f ft\n',lvd)
fprintf('   Position right of buttline = %6.1f ft\n',yvd)
fprintf('               Zero lift angle = %6.2f deg\n',alplov*57.3)
fprintf('                   Maximum Cl = %6.1f \n',clvertmax)
fprintf('        Dynamic pressure ratio = %6.2f \n',qvq)
fprintf('             Lift curve slope = %6.2f \n',av)
fprintf(' Clv change with rudder angle = %6.2f per radian\n',dclvddelr)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('        *** INPUT DATA CONTINUED (screen 8 of 8) ***')
```

```
eval(['disp(''                                 ',filename1,''')'])
disp(' ')
disp('                          Rigging')
disp(' ')
fprintf(' Long cyclic pitch/stick defl = %6.2f deg/in\n',db1mddele*57.3)
fprintf('  Lat cyclic pitch/stick defl = %6.2f deg/in\n',da1mddela*57.3)
fprintf('  Collective pitch/stick defl = %6.2f deg/in\n',dthetomddelc*57.3)
fprintf(' Tail rotor pitch change/defl = %6.2f deg/unit\n',dthetotddelp*57.3)
   disp('     Max deflection of control')
fprintf('          from neutral for NOTAR = %6.2f units\n',dphinddelp*57.3)
   disp('  Displacement of anti-torque')
fprintf('     control until full rudder = %6.2f units\n',maxr)
fprintf('       Aileron angle/stick defl = %6.2f deg/in\n',ddeladlat*57.3)
fprintf('      Elevator angle/stick defl = %6.2f deg/in\n',ddeledlong*57.3)
fprintf('       Rudder angle/stick defl = %6.2f deg/in\n',ddelrddelp*57.3)
disp(' ')
disp('press any key to continue...')
pause
    elseif choice==2, % calculated data
clc
disp(' ')
disp('          *** CALCULATED DATA (screen 1 of 2) ***')
eval(['disp(''                   ',filename1,''')'])
disp(' ')
disp('                     Main Rotor')
disp(' ')
fprintf('                          thrust = %6.1f lbs\n',T)
fprintf('                          torque = %6.1f ft-lbs\n',Qrotor)
fprintf('                   advance ratio = %6.1f \n',mu)
fprintf('       inflow parameter wrt TPP = %6.3f \n',lamp)
fprintf('                 Tip path angle = %6.1f degs\n',altpp*57.3)
fprintf('              Rotor coning angle = %6.1f degs\n',ao*57.3)
fprintf('         1st lat cyclic term-A1 = %6.1f degs\n',A1*57.3)
fprintf('        1st long cyclic term-B1 = %6.1f degs\n',B1*57.3)
fprintf('                lateral flapping = %6.2f degs\n',b1s*57.3)
fprintf('          longitudinal flapping = %6.2f degs\n',a1s*57.3)
fprintf('                    Lock number = %6.1f \n',lockno)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('          *** CALCULATED DATA (screen 2 of 2)***')
eval(['disp(''                   ',filename1,''')'])
disp(' ')
disp('               Tail Rotor (all zero if NOTAR or tiltrotor)')
disp(' ')
if tiltr==0
  fprintf('               tail rotor thrust = %6.1f lbs\n',Tt)
  fprintf('                   advance ratio = %6.1f \n',mut)
  fprintf('                inflow parameter = %6.3f \n',lampt)
  fprintf('              Rotor coning angle = %6.1f degs\n',aot*57.3)
  fprintf('                lateral flapping = %6.2f degs\n',b1st*57.3)
  fprintf('          longitudinal flapping = %6.2f degs\n',a1st*57.3)
  fprintf('                    Lock number = %6.1f \n',locknot)
  disp(' ')
end
disp(' ')
disp('press any key to continue...')
pause
```

163

```
        elseif choice==3, % state matrices
clc
disp(' ')
disp('Longitudinal uncoupled plant (A or F depending on notation)')
disp('States are [u w q theta]')
disp(' ')
disp(Flonaug)
disp(' ')
disp(' ')
disp('Longitudinal uncoupled input matrix (B or G depending on notation)')
disp('Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]')
disp(' ')
disp(Glonaug)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('Lateral/directional uncoupled plant (A or F depending on notation)')
disp('States are [v p phi r psi]')
disp(' ')
disp(Flataug)
disp(' ')
disp(' ')
disp('Lateral/directional uncoupled input matrix (B or G depending')
disp('on notation)')
disp('Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]')
disp(Glataug)
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('Coupled plant (A or F depending on notation)')
disp('States are [u w q theta v p phi r psi]')
disp(' ')
disp(Amat)
disp('press any key to continue...')
pause
clc
disp(' ')
disp('Coupled input matrix (B or G depending on notation)')
disp('Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]')
disp(' ')
disp([Bmat;0 0 0 0])
disp(' ')
disp('press any key to continue...')
pause
clc
    elseif choice==4, % eigenvalues and root loci
clc
disp(' ')
disp('After you view the root loci plot, a meta file is made.')
disp('When you are done a screen will tell you the file names')
disp('of the meta files.  To get a hard copy of the plots, you')
disp('must graphics post process (GPP) the files for your')
disp('particular printer set-up then, print.')
disp(' ')
disp('NOTE: If ALL roots are real, MATLAB will NOT plot them')
disp('      in the Argand plane, but will plot the root against')
```

164

```
disp('          its position in the vector (e.g. the first root would')
disp('          be ploted as (1,root))')
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
disp('              *** EIGENVALUES ***')
eval(['disp(''                    ',filename1,''')'])
disp(' ')
disp('Uncoupled')
disp(' ')
disp('Longitudinal plant')
disp(' ')
damp(Rlonaug)
disp(' ')
disp('Lateral/Directional plant')
damp(Rlataug)
disp(' ')
disp('press any key to continue...')
pause
plot(Rlonaug,'*'),grid,title('Roots of Longitudinal Plant')
pause
%!del rootlon.*
print -dmeta rootlon
plot(Rlataug,'*'),grid,title('Roots of Lateral/Directional Plant')
pause
%!del rootlat.*
print -dmeta rootlat
clc
disp('              *** EIGENVALUES ***')
eval(['disp(''                    ',filename1,''')'])
disp(' ')
disp('Coupled Plant')
disp(' ')
damp(Rcoup)
disp(' ')
disp('press any key to continue...')
pause
plot(Rcoup,'*'),grid,title('Roots of Coupled Plant')
pause
%!del rootcoup.*
print -dmeta rootcoup
clc
disp(' ')
disp('Plots are saved under the following filenames:')
disp(' ')
disp('Longitudinal roots - rootlon.wmf')
disp('Lateral/Directional roots - rootlat.wmf')
disp('Coupled roots - rootcoup.wmf')
disp(' ')
disp('press any key to continue...')
pause
    elseif choice==5, % key control parameters
clc
disp(' ')
   disp('       *** KEY CONTROL PARAMETERS (screen 1 of 2) ***')
eval(['disp(''                         ',filename1,''')'])
disp(' ')
fprintf('   cross coupling = %6.2f \n',xcouple)
```

165

```
   disp(' ')
   disp('                        Designed damping')
fprintf('            pitch = %6.1f ft-lbs/(rad/sec)\n',desdmdq)
fprintf('             roll = %6.1f ft-lbs/(rad/sec)\n',desdrdp)
fprintf('              yaw = %6.1f ft-lbs/(rad/sec)\n',desdndr)
   disp(' ')
   disp('                        Control Power')
fprintf('            pitch = %6.1f ft-lbs/in\n',cppitch)
fprintf('             roll = %6.1f ft-lbs/in\n',cproll)
fprintf('              yaw = %6.1f ft-lbs/in\n',cpyaw)
   disp(' ')
disp(' ')
disp('press any key to continue...')
pause
clc
disp(' ')
   disp('      *** KEY CONTROL PARAMETERS (screen 2 of 2) ***')
eval(['disp(''                         ',filename1,''')'])
disp(' ')
   disp('            Cooper Harper Pilot Ratings')
   disp('             damping/moment of inertia')
fprintf('pitch (dM/dq)/Iyy = %6.2f [ft-lbs/(rad/sec)]/(slug ft^2)\n',prpitch)
fprintf(' roll (dR/dp)/Ixx = %6.2f [ft-lbs/(rad/sec)]/(slug ft^2)\n',prroll)
fprintf('  yaw (dN/dr)/Izz = %6.2f [ft-lbs/(rad/sec)]/(slug ft^2)\n',pryaw)
   disp(' ')
   disp('            control power/moment of inertia')
fprintf('pitch (dM/in)/Iyy = %6.2f (ft-lbs/in)/(slug ft^2)\n',cpipitch)
fprintf(' roll (dR/in)/Ixx = %6.2f (ft-lbs/in)/(slug ft^2)\n',cpiroll)
fprintf('  yaw (dN/in)/Izz = %6.2f (ft-lbs/in)/(slug ft^2)\n',cpiyaw)
disp(' ')
disp(' ')
disp('press any key to continue...')
pause
    elseif choice==6,  % command bandwidth plots
clc
disp(' ')
disp('After you view a bode plot of the transfer function from')
disp('input to state output, a windows meta file is made.  When ')
disp('you exit,a screen will tell you the file names of the meta')
disp(' files. To get a hard copy of the plots, you must import ')
disp('them into your favorite word processor then print.')
disp(' ')
disp(' ')
      if Vinf<20
        cmdbwplh
      else,
        cmdbwplc
      end
    elseif choice==7,  % command time response
clc
disp(' ')
disp('After you view a time response of the transfer function from')
disp('input to state output, a windows meta file is made.  When ')
disp('you exit,a screen will tell you the file names of the meta')
disp(' files. To get a hard copy of the plots, you must import ')
disp('them into your favorite word processor then print.')
disp(' ')
disp(' ')
      if Vinf<20
        tmresph
```

166

```matlab
        else,
          tmrespc
        end
      elseif choice==0,
        flag=0;
      else
        disp(' ')
        disp('Enter a number on the menu')
        pause(3)
      end
    end
end
%
%    *** output to disk (text file) ***
%
diary off
eval(['flag=exist(''',filename1,'.stb'');']);
  if flag < 1,
    eval(['diary ',filename1,'.stb''']);
  else
    eval(['!del ',filename1,'.stb''']);
    eval(['diary ',filename1,'.stb''']);
  end
disp(' ')
disp('                        *** RESULTS ***')
eval(['disp(''                                ',filename1,''')'])
disp(' ')
disp('                      *** INPUT DATA ***')
disp(' ')
disp('                        Flight Conditions')
disp(' ')
fprintf('               Forward velocity = %6.0f kts\n',Vinf/1.69)
fprintf('                    Temperature = %6.0f degs F\n',temp)
fprintf('               Pressure altitude = %6.0f ft\n',PA)
fprintf('               Auxiliary thrust = %6.0f lbs\n',Taux)
disp(' ')
disp('                          Fuselage')
disp(' ')
fprintf('                    Gross weight = %6.0f lbs\n',GW)
fprintf('    Equivalent flat plate area = %6.1f ft^2\n',Afh)
fprintf('        Vertical projected area = %6.1f ft^2\n',Afv)
fprintf('    Center of Pressure station = %6.1f ft\n',lfd)
fprintf('    Fuselage alpha @ zero lift = %6.1f degrees\n',alplof*57.3)
fprintf(' Lift curve slope of fuselage = %6.3f 1/rad\n',af)
fprintf(' Moment coefficient @ 0 alpha = %6.3f (referenced to Aw, cw)\n',cmof)
fprintf('      Moment coeff./alpha slope = %6.3f 1/rad\n',cmalpf)
fprintf('      CG height above waterline = %6.1f ft\n',zcg)
fprintf(' Aircraft CG fuselage station = %6.1f ft\n',xcg)
fprintf('CG position right of buttline = %6.1f ft\n',ycg)
fprintf('                            Ixx = %6.1f slug ft^2\n',Ixx)
fprintf('                            Iyy = %6.1f slug ft^2\n',Iyy)
fprintf('                            Izz = %6.1f slug ft^2\n',Izz)
fprintf('                            Ixz = %6.1f slug ft^2\n',Ixz)
fprintf('                Downwash ratio = %6.2f \n',vfv1)
disp(' ')
disp('                          Main Rotor')
disp(' ')
fprintf('               Number of blades = %6.0f \n',b)
fprintf('                   Rotor radius = %6.1f ft\n',R)
if tiltr==0
```

167

```
    fprintf('              Ave Blade chord = %6.1f
ft\n',(sum(cblade)/length(cblade)))
end
fprintf('                    Blade twist = %6.2f degs\n',twist*57.3)
if Airfoil==1,
    disp('              Blade airfoil =  HH-02')
elseif Airfoil==2,
    disp('              Blade airfoil =  VR-12')
else,
    disp('              Blade airfoil =  NACA 0012')
end
fprintf('        Blade lift curve slope = %6.2f \n',a)
fprintf('                  Blade weight = %6.1f lbs\n',wblade)
fprintf('           Rotational velocity = %6.2f rads/sec\n',omega)
fprintf('             Blade grip length = %6.1f ft\n',grip)
fprintf('                   Hinge offset = %6.1f ft\n',e)
fprintf('       Flapping spring constant = %6.1f ft-lbs/deg\n',Kflpsprng/57.3)
fprintf('     Flapping moment of inertia = %6.1f slug ft^2\n',Ib)
fprintf('     Hub height above waterline = %6.1f ft\n',hmd)
fprintf('          Hub fuselage station = %6.1f ft\n',lmd)
fprintf('  Hub position rt of buttline = %6.1f ft\n',ymd)
fprintf('                Mast incidence = %6.2f deg\n',im*57.3)
disp(' ')
if notar==0
  disp('               Tail rotor (zero if NOTAR)')
  disp(' ')
  fprintf('             Number of blades = %6.1f \n',bt);
  fprintf('                   Blade chord = %6.1f ft\n',cot)
  fprintf('                  Blade radius = %6.1f ft\n',Rt)
  fprintf('              Lift curve slope = %6.2f \n',at)
  fprintf('           Rotational velocity = %6.2f rad/sec\n',ohmt)
  fprintf('     Flapping moment of inertia = %6.1f slug ft^2\n',Ibt)
  fprintf('                  Delta-3 angle = %6.2f deg\n',delta3*57.3)
  fprintf('                    Blade twist = %6.2f deg\n',theta1t*57.3);
  fprintf('     Hub height above waterline = %6.1f ft\n',htd)
  fprintf('           Hub fuselage station = %6.1f ft\n',ltd)
  fprintf('  Hub position rt of buttline = %6.1f ft\n',ytd)
  disp(' ')
elseif notar==1
  disp('                          NOTAR')
  disp(' ')
  fprintf('        Height above waterline = %6.1f ft^2\n',htnd)
  fprintf('              Fuselage station = %6.1f ft^2\n',ltnd)
  fprintf('     Position right of buttline = %6.1f ft^2\n',ytnd)
  fprintf('          NOTAR boom diameter = %6.1f ft^2\n',dian)
  fprintf('         Swirl angle at boom = %6.2f deg\n',swirl*57.3)
  fprintf('       Maximum thruster force = %6.1f lbs\n',Ytmaxn)
  fprintf('       Thrust fuselage station = %6.1f ft^2\n',lttnd)
  disp(' ')
end
disp('                          Wing')
disp(' ')
fprintf('                          Area = %6.1f ft^2\n',Swing)
fprintf('                          Span = %6.1f ft\n',bwing)
fprintf('                            CL = %6.2f \n',CLwing)
fprintf('                           CDo = %6.4f \n',CDowing)
fprintf('                      Tip cord = %6.1f ft\n',ctw)
fprintf('                     Root cord = %6.1f ft\n',crw)
fprintf('         Wing efficiency factor = %6.2f \n',ewing)
fprintf('               Zero lift angle = %6.2f deg\n',alplow*57.3)
```

168

```
fprintf('          Angle of incidence = %6.2f deg\n',iw*57.3)
fprintf('          Wing sweep angle = %6.2f deg\n',lambda*57.3)
fprintf('          Wing Dihedral angle = %6.2f deg\n',dih*57.3)
fprintf('          Lift curve slope = %6.2f per radian\n',aw)
fprintf(' Delta Clw per flaperon angle = %6.2f per radian\n',dclwddelf)
fprintf('Wing Moment Coeff @ zero lift = %6.2f \n',cmow)
fprintf('          Rotor downwash ratio = %6.2f \n',vwv1)
fprintf('          Fuselage downwash ratio = %6.2f \n',detafdalpfw)
fprintf('  Downwash angle @ zero alpha = %6.2f radians\n',epso)
fprintf('          Height above waterline = %6.1f ft\n',hwd)
fprintf('Fuselage station (of wing CP) = %6.1f ft\n',lwd)
fprintf('CP Position right of buttline = %6.1f ft\n',ywd)
disp(' ')
disp('                    Horizontal tail')
disp(' ')
fprintf('                        Area = %6.1f ft^2\n',Shoriz)
fprintf('                        Span = %6.1f ft\n',bhoriz)
fprintf('                          CL = %6.2f \n',CLhoriz)
fprintf('                         CDo = %6.4f \n',CDohoriz)
fprintf('          Zero lift angle = %6.2f deg\n',alploh*57.3)
fprintf('          Angle of incidence = %6.2f deg\n',iw*57.3)
fprintf('          Lift curve slope = %6.1f \n',ah)
fprintf('          Height above waterline = %6.1f ft\n',hhd)
fprintf('          Fuselage station = %6.1f ft\n',lhd)
fprintf('    Position right of buttline = %6.1f ft\n',yhd)
fprintf('          Dynamic pressure ratio = %6.2f \n',qhq)
fprintf('          Rotor downwash ratio = %6.2f \n',vhv1)
fprintf('          Fuselage downwash ratio = %6.2f \n',depsdalph)
fprintf('    H-Tail Span Efficiency (e) = %6.2f \n',1/(1+delih))
fprintf('Delta Clh per elevator angle  = %6.2f per radian\n',dclhddeleh)
disp(' ')
disp('                    Vertical tail')
disp(' ')
fprintf('                        Area = %6.1f ft^2\n',Svert)
fprintf('                        Span = %6.1f ft\n',bvert)
fprintf('                          CL = %6.2f \n',CLvert)
fprintf('                         CDo = %6.4f \n',CDovert)
fprintf('          Height above waterline = %6.1f ft\n',hvd)
fprintf('          Fuselage station = %6.1f ft\n',lvd)
fprintf('    Position right of buttline = %6.1f ft\n',yvd)
fprintf('          Zero lift angle = %6.2f deg\n',alplov*57.3)
fprintf('          Maximum Cl = %6.2f \n',clvertmax)
fprintf('          Dynamic pressure ratio = %6.2f \n',qvq)
fprintf('          Lift curve slope = %6.2f \n',av)
fprintf(' Clv change with rudder angle = %6.2f per radian\n',dclvddelr)
disp(' ')
disp('                    Rigging')
disp(' ')
fprintf('   Long cyclic pitch/inch defl = %6.2f deg/in\n',db1mddele*57.3)
fprintf('    Lat cyclic pitch/inch defl = %6.2f deg/in\n',da1mddela*57.3)
fprintf('    Collective pitch/inch defl = %6.2f deg/in\n',dthetomddelc*57.3)
fprintf(' Tail rotor pitch change/defl = %6.2f deg/unit\n',dthetotddelp*57.3)
disp('     Max deflection of control')
fprintf('          from neutral for NOTAR = %6.2f units\n',dphinddelp*57.3)
disp('  Displacement of anti-torque')
fprintf('     control until full rudder = %6.2f units\n',maxr)
if tiltr==1
   fprintf('    Aileron angle/stick defl = %6.2f deg/in\n',ddeladlat*57.3)
   fprintf('      Coll. pitch/stick defl = %6.2f deg/in\n',dthetomddelc*57.3)
   fprintf('   Elevator angle/stick defl = %6.2f deg/in\n',ddeledlong*57.3)
```

169

```
    fprintf('     Rudder angle/stick defl = %6.2f deg/in\n',ddelrddelp*57.3)
end
disp(' ')
disp('                    *** CALCULATED DATA ***')
disp(' ')


disp(' ')
disp('                          State Matrices')
disp(' ')
disp('Longitudinal uncoupled plant (A or F depending on notation)')
disp('States are [u w q theta]')
disp(' ')
disp(Flonaug)
disp(' ')
disp(' ')
disp('Longitudinal uncoupled input matrix (B or G depending on notation)')
disp('Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]')
disp(' ')
disp(Glonaug)
disp(' ')
disp(' ')
disp('Lateral/directional uncoupled plant (A or F depending on notation)')
disp('States are [v p phi r psi]')
disp(' ')
disp(Flataug)
disp(' ')
disp(' ')
disp('Lateral/directional uncoupled input matrix (B or G depending')
disp('on notation)')
disp('Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]')
disp(Glataug)
disp(' ')
disp(' ')
disp('Coupled plant (A or F depending on notation)')
disp('States are [u w q theta v p phi r psi]')
disp(' ')
disp(Amat)
disp(' ')
disp(' ')
disp('Coupled input matrix (B or G depending on notation)')
disp('Inputs are [longitudinal cyclic, collective, lateral cyclic, pedals]')
disp(' ')
disp(Bmat)
disp(' ')
disp(' ')
disp('                          Eigenvalues')
disp(' ')
disp('Uncoupled')
disp(' ')
disp('Longitudinal plant')
disp(' ')
damp(Rlonaug);
disp(' ')
disp('Lateral/Directional plant')
disp(' ')
damp(Rlataug)
disp(' ')
disp(' ')
disp('Coupled Plant')
disp(' ')
```

```
damp(Rcoup)
disp(' ')
disp(' ')
disp('          *** KEY CONTROL PARAMETERS ***')
disp(' ')
%fprintf('   cross coupling = %6.2f \n',xcouple)
   disp(' ')
   disp('                      Designed damping')
fprintf('          pitch = %6.1f ft-lbs/(rad/sec)\n',desdmdq)
fprintf('           roll = %6.1f ft-lbs/(rad/sec)\n',desdrdp)
fprintf('            yaw = %6.1f ft-lbs/(rad/sec)\n',desdndr)
   disp(' ')
   disp('                      Control Power')
fprintf('          pitch = %6.1f ft-lbs/in\n',cppitch)
fprintf('           roll = %6.1f ft-lbs/in\n',cproll)
fprintf('            yaw = %6.1f ft-lbs/in\n',cpyaw)
   disp(' ')
   disp('             Cooper Harper Pilot Ratings')
   disp('               damping/moment of inertia')
fprintf('pitch (dM/dq)/Iyy = %6.2f [ft-lbs/(rad/sec)]/(slug ft^2)\n',prpitch)
fprintf(' roll (dR/dp)/Ixx = %6.2f [ft-lbs/(rad/sec)]/(slug ft^2)\n',prroll)
fprintf('  yaw (dN/dr)/Izz = %6.2f [ft-lbs/(rad/sec)]/(slug ft^2)\n',pryaw)
   disp(' ')
   disp('            control power/moment of inertia')
fprintf('pitch (dM/in)/Iyy = %6.2f (ft-lbs/in)/(slug ft^2)\n',cpipitch)
fprintf(' roll (dR/in)/Ixx = %6.2f (ft-lbs/in)/(slug ft^2)\n',cpiroll)
fprintf('  yaw (dN/in)/Izz = %6.2f (ft-lbs/in)/(slug ft^2)\n',cpiyaw)
disp(' ')
disp(' ')
diary off
%
clc
disp(' ')
disp('     *** OUTPUT DATA INSTRUCTIONS (screen 1 of 3) ***')
disp(' ')
disp('Because this subroutine generates a large number of single')
disp('value data not shown on the output screen, a text file')
disp('VARLIST.TXT is on this disk which lists the variable names')
disp('for all the stability derivatives.  Stability derivative')
disp('contributions for all major aircraft components can be found')
disp('by reading the text file VARLIST.TXT, then asking MATLAB the')
disp('variable name corresponding to the derivative.')
disp(' ')
disp('Press any key to continue')
pause
 clc
disp(' ')
disp('     *** OUTPUT DATA INSTRUCTIONS (screen 2 of 3) ***')
disp(' ')
eval(['disp(''A. Data from the output screen saved to a file named:          \
',filename1,'.stb'')'])
disp('   This is a text file, use the TYPE command to view the file')
disp('   or use a text editor to view/print the file.')
disp(' ')
disp('B. Matrix and vector data saved to a default file named: mstabdat.mat')
disp('   This is a ".mat" binary file, use the LOAD command to')
disp('   retrieve the data for plotting.')
disp(' ')
disp('C. Rename "mstabdat.mat" to another ".mat" file.')
disp('   The file "mstabdat.mat" will be overwritten when')
```

171

```
disp('   the program is executed.')
disp(' ')
eval(['disp(''D. Do not rename the file as "',filename1,'.mat"'')'])
eval(['disp(''   The file "',filename1,'.mat" is already on disk'')'])
disp('   and used for future editing.')
disp(' ')
disp('Press any key to continue')
pause
%
%    *** Output to disk (.mat file containing matrix variables
%    Amat Bmat Rcoup Flataug Glataug Rlataug Plataug Flonaug
%    Glonaug Rlonaug Plonaug
%
%    *** Configuring variables for output ***
%
save mstabdat Amat Bmat Rcoup Flataug Glataug Rlataug Plataug Flonaug ...
 Glonaug Rlonaug Plonaug
%
clc
disp(' ')
disp('      *** OUTPUT DATA INSTRUCTIONS (screen 3 of 3) ***')
disp(' ')
disp('A. Single value data saved to a default file named: vstabdat.mat')
disp('   This is a ".mat" binary file, use the LOAD command to')
disp('   retrieve the data for plotting.')
disp(' ')
disp('B. Rename "vstabdat.mat" to another ".mat" file.')
disp('   The file "vstabdat.mat" will be overwritten when')
disp('   the program is executed.')
disp(' ')
eval(['disp(''C. Do not rename the file as "',filename1,'.mat"'')'])
eval(['disp(''   The file "',filename1,'.mat" is already on disk'')'])
disp('   and used for future editing.')
%
%    *** Configuring variables for output ***
%
clear Amat Bmat Rcoup Flataug Glataug Rlataug Plataug Flataug ...
 Glataug Rlonaug Plonaug num den vA1 vB1 va1s vb1s vmu vtheta7 ...
 vao vv1 vlamp vthetao vctsig vcqsig vchsig valtpp
%
save vstabdat
clear
clc
disp(' ')
disp( '                *** END STABILITY AND CONTROL ROUTINE ***')
disp(' ')
disp('press any key to continue...')
pause
format loose
%
% return to JANRAD.M
```

172

```
% CMDBWPLH.M
% open loop response plots for longitudinal and lateral plants %
disp('While viewing a plot, press any key to go to the next plot')
disp(' ')
disp('Do you want to see longitudinal or lateral/directional plots?')
disp(' ')
disp('1.  Longitudinal (eight plots total).')
disp('2.  Lateral Directional (ten plots total).')
disp(' ')
pview=input('Enter a number : ');
clc
w=logspace(-2,2);
if pview==1
% open loop response plots for longitudinal and lateral plants
w=logspace(-2,2);
Du=[0 0 0 0];
Cu=[1 0 0 0];
Cthet=[0 0 0 1];
Cqrat=[0 0 1 0];
Cw=[0 1 0 0];
disp('longitudinal cylic')
% command bw e to u
[NUM,DEN]=ss2tf(Flonaug,Glonaug,Cu,Du,1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Longitudinal Cyclic to U, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbe2uh
% command bw e to theta
[NUM,DEN]=ss2tf(Flonaug,Glonaug,Cthet,Du,1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Longitudinal Cyclic to Theta, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbe2theh
% command bw e to q
[NUM,DEN]=ss2tf(Flonaug,Glonaug,Cqrat,Du,1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Longitudinal Cyclic to Pitch Rate, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbe2qh
% command bw e to w
[NUM,DEN]=ss2tf(Flonaug,Glonaug,Cw,Du,1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Longitudinal Cyclic to W, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbe2wh
%
% now collective
disp('collective')
% command bw c to u
[NUM,DEN]=ss2tf(Flonaug,Glonaug,Cu,Du,2);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Collective to U, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbc2uh
% command bw c to theta
```

173

```
 [NUM,DEN]=ss2tf(Flonaug,Glonaug,Cthet,Du,2);
 semilogx(w,20*log10(bode(NUM,DEN,w))),grid
 title('Open loop response Collective to Pitch, Hover')
 xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
 pause
 print -dmeta cbc2theh
 % command bw c to q
 [NUM,DEN]=ss2tf(Flonaug,Glonaug,Cqrat,Du,2);
 semilogx(w,20*log10(bode(NUM,DEN,w))),grid
 title('Open loop response Collective to Pitch Rate, Hover')
 xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
 pause
 print -dmeta cbc2qh
 % command bw c to w
 [NUM,DEN]=ss2tf(Flonaug,Glonaug,Cw,Du,2);
 semilogx(w,20*log10(bode(NUM,DEN,w))),grid
 title('Open loop response Collective to W, Hover')
 xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
 pause
 print -dmeta cbc2wh
 clc
 disp(' ')
 disp('Plots are saved under the following filenames:')
 disp(' ')
 disp('Longitudinal Cyclic')
 disp('Longitudinal Cyclic to U, Hover - cbe2uh.wmf')
 disp('Longitudinal Cyclic to Theta, Hover - cbe2theh.wmf')
 disp('Longitudinal Cyclic to Pitch Rate, Hover - cbe2qh.wmf')
 disp('Longitudinal Cyclic to W, Hover - cbe2wh.wmf')
 disp('')
 disp('Collective')
 disp('Collective to U, Hover - cbc2uh.wmf')
 disp('Collective to Pitch, Hover - cbc2theh.wmf')
 disp('Collective to Pitch Rate, Hover - cbc2qh.wmf')
 disp('Collective to W, Hover - cbc2wh.wmf')
 disp(' ')
 disp('press any key to continue ...')
 pause
 %
 % now for lateral directional plant
 elseif pview==2
 %
 % now for lateral directional plant
 Du=[0 0 0 0];
 Cphi=[0 0 1 0 0];
 Cv=[1 0 0 0 0];
 Cp=[0 1 0 0 0];
 Cr=[0 0 0 1 0];% yaw rate
 Cy=[0 0 0 0 1];% yaw angle
 % lateral cyclic
 disp('lateral cyclic')
 % command bw lateral cyclic to bank
 [NUM,DEN]=ss2tf(Flataug,Glataug,Cphi,Du,3);
 semilogx(w,20*log10(bode(NUM,DEN,w))),grid
 title('Open loop response Lateral Cyclic to Bank, Hover')
 xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
 pause
 print -dmeta cba2phih
 % command bw lateral cyclic to sideslip (lateral velocity)
 [NUM,DEN]=ss2tf(Flataug,Glataug,Cv,Du,3);
```

174

```
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Lateral Cyclic to Sideslip (v), Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cba2vh
% command bw lateral cyclic to roll rate
[NUM,DEN]=ss2tf(Flataug,Glataug,Cp,Du,3);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Lateral Cyclic to Roll Rate, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cba2ph
% command bw lateral cyclic to yaw rate
[NUM,DEN]=ss2tf(Flataug,Glataug,Cr,Du,3);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Lateral Cyclic to Yaw Rate, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cba2rh
% command bw lateral cyclic to yaw angle
[NUM,DEN]=ss2tf(Flataug,Glataug,Cy,Du,3);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Lateral Cyclic to Yaw, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cba2yh
%
% pedals
% command bw pedals to bank
disp('pedals')
[NUM,DEN]=ss2tf(Flataug,Glataug(:,4)*30,Cphi,[0],1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Pedals to Bank, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbp2phih
% command bw pedals to sideslip
[NUM,DEN]=ss2tf(Flataug,Glataug(:,4)*30,Cv,[0],1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Pedals to Sideslip (v), Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbp2vh
% command bw pedals to roll rate
[NUM,DEN]=ss2tf(Flataug,Glataug(:,4)*30,Cp,[0],1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Pedals to Roll Rate, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbp2ph
% command bw pedals to yaw rate
[NUM,DEN]=ss2tf(Flataug,Glataug(:,4)*30,Cr,[0],1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
title('Open loop response Pedals to Yaw Rate, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbp2rh
% command bw pedals to yaw
[NUM,DEN]=ss2tf(Flataug,Glataug(:,4)*30,Cy,[0],1);
semilogx(w,20*log10(bode(NUM,DEN,w))),grid
```

```
title('Open loop response Pedals to Yaw, Hover')
xlabel('Angular Frequency (rad/sec)'),ylabel('Gain (dB)')
pause
print -dmeta cbp2yh
clc
disp(' ')
disp('Plots are saved under the following filenames:')
disp(' ')
disp('Lateral cyclic')
disp('Lateral Cyclic to Bank, Hover - cba2phih.wmf')
disp('Lateral Cyclic to Sideslip (v), Hover - cba2vh.wmf')
disp('Lateral Cyclic to Roll Rate, Hover - cba2ph.wmf')
disp('Lateral Cyclic to Yaw Rate, Hover - cba2rh.wmf')
disp('Lateral Cyclic to Yaw, Hover - cba2yh.wmf')
disp('Pedals')
disp('Pedals to Bank, Hover - cbp2phih.wmf')
disp('Pedals to Sideslip (v), Hover - cbp2vh.wmf')
disp('Pedals to Roll Rate, Hover - cbp2ph.wmf')
disp('Pedals to Yaw Rate, Hover - cbp2rh.wmf')
disp('Pedals to Yaw, Hover - cbp2yh.wmf')
disp(' ')
disp('press any key to continue ...')
pause
%
else
end
```

APPENDIX M. ADDED JANRAD SCRIPT (MATLAB *.M) FILES

CONTENTS

```
%%%%%%%%%%%%%%%%%%%%%
%       aptrim.m
%
%       Routine called by CTLTRGRP.m
%
%       calculates the static stability in airplane configuration
%
%       *** Wing ***
%
lw = lwd - xcg;                     % a.c. long. offset
hw = hwd - zcg;                     % a.c. vert. offset
Aw = Swing;
bw = bwing;
cdow = CDowing;
deliw = 1/ewing - 1;
qwq = vwv1^2;
cw = (crw+ctw)/2;
delf = 0;                  % ****NEEDS TO BE INCORPORATED***
%
%       *** Horiz. Stab. ***
%
lh = lhd - xcg;                     % long. h. stab. offset (ft)
hh = hhd - zcg;                     % vertical h. stab offset (ft)
Ah = Shoriz;
bh = bhoriz;
cdoh = CDohoriz;
qhq = vhv1^2;
Vh = (lh.*Ah)./(cw.*Aw);    % Hor. Tail Volume Ratio
%
%       *** Vertical Stab. ***
%
hv = hvd - zcg;                     % Vertical stab. offset (ft)
lv = lvd - xcg;                     % long. v. stab. offset (ft)
Av = Svert;
cdov = CDovert;
qvq = 1;
Vv = (lv.*Av)./(cw.*Aw);    % Ver. Tail Volume Ratio
%dclvddelp=flapchor(tch,cfcv);    % curve fit function to calc.
                          %    change of lift wrt elevator
%
%       *** Aircraft ***
%
Talt=518.69-.00356*PA;
rho=.002377*(Talt/518.69)^(32.174/1716/((518.69-390.53)/36000)-1);
q=0.5*rho*Vinf^2;
Ic = Ixx*Izz - Ixz^2;
lfd = 293/12;
lf = lfd - xcg;
gamc = 0;                % *** NEED TO
hn = acw + Vh.*ah./aw*(1-depsdalph);
LSM = hn - (xcg-lwd)/cw + acw;
%
%       Calculations required for CTLTRGRP.M
%
%
%       *** Trim ***
%
if ih == []   % All movable horiz. stab (stabilator)
%
  ih = zeros(size(Vinf));
```

178

```
      dLdalp = q*Aw*awb + qhq*q*ah*Ah.*(1-depsdalph);
      dLdih = qhq*q*ah*Ah;
      Ltr = GW + q.*(Aw.*(awb*alplow - dclwddelf.*delf) - qhq*ah*Ah*...      % #1
                   (ih - iw - depsdalph.*(dclwddelf.*delf/aw - alplow)));
      dMdalp = -lw*q*Aw*awb - lh*qhq*q*ah*Ah.*(1-depsdalph);
      dMdih = -lh*qhq*q*ah*Ah;
      Mtr = -q.*(Aw.*(cmowb*cw + lw*(awb*alplow - dclwddelf.*delf)) + ...
              lh*qhq*ah*Ah*(iw + depsdalph.*(dclwddelf.*delf/aw - alplow)));
      Trim = [dLdalp dLdih;dMdalp dMdih]\[Ltr;Mtr];
      alpw = Trim(1);
      ih = Trim(2);
%
    else%,disp('          % Conventional elevator')
%
      dLdalp = q*(Aw*(aw+af) + qhq*ah*Ah*(1-depsdalph));
      dLddele = q*qhq*Ah*dclhddeleh;
      Ltr = GW + q*(Aw*(aw*alplow + af*(iw+alplof)) - qhq*Ah*ah*...
              (ih-iw-epso));
      dMdalp = -q*(Aw*(lw*aw+lf*af-cw*cmalpf)+lh*qhq*ah*Ah*(1-depsdalph));
      dMddele = -lh*qhq*q*Ah*dclhddeleh;
      Mtr = q*(Aw*(cw*(cmalpf*iw-cmow-cmof)-lw*aw*alplow)-...
              lf*af*(iw+alplof)+lh*qhq*Ah*ah*(ih-iw-epso));
      Trim = [dLdalp dLddele;dMdalp dMddele]\[Ltr;Mtr];
      alpw = Trim(1);
      dele = Trim(2);
    end
    tho = alpw + gamc - iw;
    uo = Vinf.*cos(tho-gamc);
    vo = 0;
    wo = -Vinf.*sin(tho-gamc);
%
%         *** Wing ***
%
    Clw = aw.*(alpw-alplow);
    Lw = qwq.*q.*Aw.*Clw;
    Dw = qwq.*q.*Aw.*(Aw.*Clw.^2.*(1+deliw)/pi/bw^2 + cdow);
    Xw = Lw.*sin(tho-gamc)-Dw.*cos(tho-gamc);
    Zw = -Lw.*cos(tho-gamc)-Dw.*sin(tho-gamc);
%
%         *** Fuselage ***
%
    Df = q*Afh;
    Lf = q*Aw*af*(tho-alplof);
    Xf = -Df.*cos(tho-gamc) + Lf.*sin(tho-gamc);
    Yf = 0;
    Zf = -Lf.*cos(tho-gamc)-Df.*sin(tho-gamc);
    dmqdalpf=53.74;
    Mf = q.*dmqdalpf.*tho;
    Nf = 0;              %q*Nqf;
    Rf = 0;              %q*Rqf;
%
%         *** Horizontal Stabilizer ***
%
    alph = tho+ih-gamc-epso-depsdalph.*alpw;
    Clh = ah.*alph + dclhddeleh*dele;
    Lh = qhq.*q.*Ah.*Clh;
    Dh = qhq.*q.*Ah.*(Ah.*Clh.^2.*(1+delih)/pi/bh^2 + cdoh);
    Xh = Lh.*sin(alph-ih-gamc) - Dh.*cos(alph-ih-gamc);
    Zh = -Lh.*cos(alph-ih-gamc) - Dh.*sin(alph-ih-gamc);
%
```

179

```
%       *** Vertical Stabilizer ***
%
Dv = qvq.*q.*Av.*cdov;
Xv = -Dv.*cos(tho-gamc);
pho=0;
%
%       *** Protors ***
%
Xp = -(Xh + Xf + Xv + Xw) + GW*sin(tho);
T=Xp;
Qrotor = 6989;
mu=0;
lamp=0;
altpp=0;
ao=0;
A1=0;
B1=0;
b1s=0;
a1s=0;
lockno = (rho*a*rchord*R^4)/Ib;
Tt=0;
mut=0;
lampt=0;
aot=0;
b1st=0;
a1st=0;
locknot=0;
hm = hmd-zcg;
```

```
% CTLTRGRP.M
% CALLED BY TLTRCRUS.M
% Computes the stability derivatives of the fuselage, wing, vertical
% fin and horizontal stabalizer in cruise flight.
% Uses data loaded in the workspace by JANRAD.M STAB.M TLTRCRUS.M
%
% THIS SUBROUTINE MUST FOLLOW APTRIM.M
%
% Compute the stability derivatives of the FUSELAGE for tilt-rotor
%        forward flight (airplane mode)
%
%dclhddeleh = ah;                    % No Elevator (dele = ih)
dxdxdotm = -154.3;                   % Due to Protor Thrust
dmdxdotm = -dxdxdotm*hw;
dgamdzdotf=-1/Vinf;
dbetadydotf=1/Vinf;
dalpfdzdotf=-dgamdzdotf;
%
%                                    *** NEED TO MODIFY ***
%
depsfdalpfw=-0.1;                    %        ALL
dfdalpf = 98.48*(tho-gamc);          %       XV-15
dlqdalpf = af*Aw;                    %       DATA
dsfqdbetaf = -1.45*57.3;             %     EXTRACTED
dmqdalpf = cmalpf*Aw*cw;             %       FROM
dnqdbetaf = -20.2*57.3;              %       GTRS
drqdbetaf = -7.5*57.3;               %   DOCUMENTATION
%
dxdxdotf=2*Xf/Vinf;
dxdzdotf=(Lf-q*dfdalpf)*dalpfdzdotf;
dxdthetom = 2020/dthetomddelc;                       %49.14*(Vinf/1.68894) + 17582;
dydydotf=1/Vinf*(q*dsfqdbetaf-Df);
dzdzdotf=-(Df+q*dlqdalpf)*dalpfdzdotf;
dzdxdotf=2*Zf/Vinf;
dzdthetom=0;
drdydotf=q*drqdbetaf*dbetadydotf;
dmdxdotf=2*Mf/Vinf;
dmdzdotf=q*dmqdalpf*dalpfdzdotf;
dmdthetom=-dxdthetom*hw;
dndydotf=q*dnqdbetaf*dbetadydotf;
%
% Compute the stability derivatives of the WING
%
%      ** Flap relations can be incorporated later **
delf=0;                                  %dclwddelf=0;
alplow = alplow - dclwddelf*delf/aw;         %  effective one due to flap
  dgamdzdotw=-1/Vinf;
  depsfwdzdotw=depsfdalpfw*(-dgamdzdotw);    %1/(4*q*pi*R^2)*dzdzdotm
  dalphdzdotw=-(depsfwdzdotw+dgamdzdotw);
%
  dxdxdotw=2*Xw/Vinf;
  dxdzdotw=qwq*q*Aw*aw*((alpw-alplow)*(1-2*aw*(1+deliw)*Aw/pi/bw^2)+alpw-iw)*...
    dalphdzdotw;
  dxdrw=0;
  dxddela=0;
  dxddelr=0;
  dyddela = 0;
  dzdxdotw=2*Zw/Vinf;
  dzdydotw=0;
  dzdzdotw=-q*Aw*aw(1+aw*(1+deliw)*Aw/pi/bw^2*(2*(alpw-alplow)*...
```

181

```
        (alpw-iw)+(alpw*alplow)^2)+cdow)*dalphdzdotw;
%
    dzdpw=0;
    dzdrw=0;
    dzdqw=dzdzdotw*lw;
    dmdxdotw=-dxdxdotw*hw+dzdxdotw*lw;
    dmdzdotw=-dxdzdotw*hw+dzdzdotw*lw;
    dmdqw=dzdqw*lh;
    dmddela=0;
    %
    drdrw = -dzdxdotw*ywd^2;                         %-dzdxdotw*(bw/3)^2
    drdpw = dzdzdotw*ywd^2;                          %dzdzdotw*(bw/3)^2
    drddela = q*Aw*bw*dclwddelf;                     %q*Aw*bw/4*dclwddelf
    dndpw = -dxdzdotw*ywd^2;                         %-dxdzdotw*(bw/3)^2
    dndrw = dxdxdotw*ywd^2;                          %dxdxdotw*(bw/3)^2
    dnddela = -0.5*q*Aw^2*(1+deliw)*dclwddelf*Clw/pi/bw;
%
% Compute the stability derivatives of the VERTICAL FIN
%
depsfdbeta=.06; % assumed to be .06 because of little study of effect
dbetadydotv=1/Vinf;
depsfdydotv=depsfdbeta*dbetadydotv;
dalpvdydotv=-(dbetadydotv+depsfdydotv);
%
dxdxdotv=2*Xv/Vinf;
dxdydotv=0;
dydxdotv=0;
dydydotv=qvq*q*Av*av*dalpvdydotv;
%
dydpv=dydydotv*hv;
dydrv=-dydydotv*lv;
drdxdotv=dydxdotv*hv;
drdydotv=dydydotv*hv;
drdpv=dydpv*hv;
drdrv=dydrv*hv;
dndxdotv=0;
dndydotv=-dydydotv*lv;
dndpv=-dydpv*lv;
dndrv=-dydrv*lv;
%
dyddelr = -q*qvq*Av*dclvddelr;
drddelr = dyddelr*hv;
dnddelr = -dyddelr*lv;
%
% zero out tail rotor derivatives:
%
    dydxdott=0;dydydott=0;dydpt=0;dydrt=0;dydthetot=0;drdxdott=0;
    drdydott=0;drdpt=0;drdrt=0;drdthetot=0;dndxdott=0;dndydott=0;
    dndpt=0;dndrt=0;dndthetot=0;
    mut=0;lampt=0;aot=0;b1st=0;a1st=0;locknot=0;
    drdthetot=0;dndthetot=0;dydthetot=0;
%
% zero out NOTAR derivatives:
%
    dydxdotn=0;dydydotn=0;dydzdotn=0;dydpn=0;dydrn=0;drdxdotn=0;
    drdydotn=0;drdzdotn=0;drdpn=0;drdrn=0;dndxdotn=0;dndydotn=0;
    dndzdotn=0;dndpn=0;dndrn=0;
    dndphin=0;drdphin=0;dydphin=0;
    dnddelv=0;drddelv=0;dyddelv=0;
%
```

182

```
% Compute the stability derivatives of the HORIZONTAL STABILIZER
%
dgamdzdoth=-1/Vinf;
depsfhdzdoth=depsdalph*(-dgamdzdoth);              %1/(4*q*pi*R^2)*dzdzdotm
dalphdzdoth=-(depsfhdzdoth+dgamdzdoth);
%
dxdxdoth=2*Xh/Vinf;
dxdzdoth=qhq*q*Ah*ah*((alph-alploh)*(1-2*ah*(1+delih)*Ah/pi/bh^2)+alph-ih)*...
  dalphdzdoth;
dxddele=0;                                 % Needs to be Nonzero
dyddele=0;
dzdxdoth=2/Vinf*Zh;
dzdzdoth=-qhq*q*Ah*ah*(1+ah*(1+delih)*Ah/pi/bh^2*(2*(alph-alploh)*...
  (alph-ih))+(alph*alploh)^2+cdoh)*dalphdzdoth;
%
dzdqh=dzdzdoth*lh;
dzddele=-qhq*q*Ah*dclhddeleh;
dmdxdoth=-dxdxdoth*hh+dzdxdoth*lh;
dmdzdoth=-dxdzdoth*hh+dzdzdoth*lh;
dmdqh=dzdqh*lh;
dmddele=-qhq*q*Ah*lh*dclhddeleh;
%
% return to TLTRCRUS.M
```

183

```
% HTLTRGRP.M
% CALLED BY TLTRHOVR.M
% Computes the basic tilt-rotor derivatives at a hover
% Computes the stability derivatives of the tiltrotor at a hover
% Uses data loaded in the workspace by JANRAD.M and STAB.M
%
% Compute the BASIC tilt-rotor derivatives at a hover
%
dmudxdot=1/omega/R;
dlampdzdot=dmudxdot;
da1dmu=8/3*thetao+2*theta1-2*v1/omega/R;
db1dmu=4/3*ao;
dctsigdlamp=inv(8/a+(sqrt(solidity/2)/(sqrt(ctsig))));
dcqsigdlamp=-a/4*(theta75-2*v1/omega/R);
da1dq=-(16/(lockno*omega*(1-e/R)^2))-12*e/R/(lockno*omega*(1-e/R)^3);
db1dp=da1dq;
da1dp=1/omega*(1-(192*e/R/(lockno^2*(1-e/R)^5)));
db1dq=-da1dp;
da1da=12*e/R/(lockno*(1-e/R)^3);
db1db=da1da;
da1db=-1/(1+((144*(e/R)^2)/(lockno^2*(1-e/R)^6)));
db1da=-da1db;
dchsigda=3/2*ctsig*(1-a/18*theta75/ctsig);
dmudlamp = 1/(B1+a1s);
dchsigdlamp = dchsigda*da1dmu*dmudlamp;
dcysigdb=dchsigda;
dmda1s=3/4*e/R*Ab*rho*R*(omega*R)^2*a/lockno+Kflpsprng;
drdb1s=dmda1s;
%
% Compute the tilt-rotor stability derivatives at a hover
%
dxdxdotm = 2*(-rho*Ab*(omega*R)^2*dchsigda*da1dmu*dmudxdot);
dxdydotm = 0;          % -rho*Ab*(omega*R)^2*dcysigdb*db1dmu*dmudxdot;
dxdzdotm = -2*rho*Ab*(omega*R)^2*(a1s+im)*dctsigdlamp*dlampdzdot;
dxdqm = -2*rho*Ab*(omega*R)^2*dchsigda*da1dq-dxdxdotm*hm;
dxdpm = 0;             % -rho*Ab*(omega*R)^2*dchsigda*da1dp
dxdrm = 0;             % -dxdxdotm*ym
dxdthetom = 2*(-rho*Ab*(omega*R)^2*(a1s+im)*dctsigdtheto);
dxda1m = -rho*Ab*(omega*R)^2*dchsigda*da1da;          % Evaluated PER ROTOR
                                                      %  for future der. only
dxdb1m = 2*(-rho*Ab*(omega*R)^2*dchsigda*da1db);
dydxdotm = 0;          % (rho*Ab*(omega*R)^2*dcysigdb*db1dmu*dmudxdot)
dydydotm = dxdxdotm;
dzdzdotm = -2*rho*Ab*(omega*R)^2*dctsigdlamp*dlampdzdot;
dydzdotm = 0;                                      % dzdzdotm*b1s
dydqm = 0;             % dxdpm
dydrm = -2*(rho*Ab*(omega*R)^2*dcysigdb*db1dmu*dmudxdot)*ym;       % should be 0
dydpm = -dxdqm+dydydotm*hm;                   % FIX THIS
dydthetom = 0;                     %(rho*Ab*(omega*R)^2*(2*(.015))*dctsigdtheto);
                   % need to solve for      b1s
dyda1m = -2*(rho*Ab*(omega*R)^2*b1s*dctsigdtheto);     % dxdb1m; Actually
dyddelc
                                              % on each rotor (diff. coll.)
dydb1m = rho*Ab*(omega*R)^2*dcysigdb*db1db;             % -dxda1m PER ROTOR
                                              % for future der. only
dydthetot = -2*rho*Ab*(omega*R)^2*dcysigdb*db1db;     % Actually -2*|dydb1m|
                                              % (diff. long. cyclic)
dzdxdotm = 0;          %  FIX THIS
dzdydotm = 0;
```

184

```
dzdpm = 0;
dzdqm = 0;                % FIX THIS
dzdrm = 0;
dzdthetom = -2*rho*Ab*(omega*R)^2*dctsigdtheto;
dzda1m = 0;
dzdb1m = 0;
drdxdotm = 0;             % drdb1s*db1dmu*dmudxdot+dydxdotm*hm
drdydotm = dydydotm*hm - 2*dmda1s*da1dmu*dmudxdot;
drdzdotm = 0;
drdqm = 0; %drdb1s*db1dq+dydqm*hm
drdrm = 2*(rho*Ab*(omega*R)^2*dcysigdb*db1dmu*dmudxdot*hm)*ym;
drdpm = dzdzdotm*ym^2+dydpm*hm;                    %drdb1s*db1dp+dydpm*hm
drdthetom = 0;                                     %dydthetom*hm+dzdthetom*ym
drda1m = 2*drdb1s*db1da + dyda1m*hm - dzdthetom*ym;  % dxdb1m; Actually drddelc
                                          % (diff. collective)
drdb1m = 0;                                       %drdb1s*db1db+dydb1m*hm;


drdthetot = 2*(drdb1s*db1db + dydb1m*hm + dzdb1m*ym); % Actually drdb1
                                       %   (diff. long. cyclic)
dmdxdotm = 2*dmda1s*da1dmu*dmudxdot-dxdxdotm*hm;
dmdydotm = 0;           %-dxdydotm*hm+dzdydotm*lm
dmdzdotm = -dxdzdotm*hm+dzdzdotm*lm;
dmdqm = 2*dmda1s*da1dq-dxdqm*hm;
dmdrm = 0;                                %-dxdrm*hm+dzdrm*lm
dmdpm = 0;                                % (dmda1s*da1dp-dxdpm*hm+dzdpm*lm)
dmdthetom = -dxdthetom*hm + dzdthetom*lm;
dmda1m = 0;                               % (dmda1s*da1da-dxda1m*hm+dzda1m*lm)
dmdb1m = 2*dmda1s*da1db - dxdb1m*hm;
dndxdotm = 0;                                  %-dxdxdotm*ym-dydxdotm*lm
dndzdotm = 0;                             % (-dydzdotm*lm-dxdzdotm*ym)
dndydotm = 2*rho*Ab*(omega*R)^2*dcysigdb*db1dmu*dmudxdot*ym-dydydotm*lm;
dndqm = 0;                                % -dxdqm*ym-dydqm*lm
dndpm = (2*rho*Ab*(omega*R)^2*dchsigda*da1dp-dxdzdotm*ym)*ym-dydpm*lm;
dndrm = (-dydrm*lm+dxdxdotm*ym)*ym;       %
dndthetom = 0;                            % (-dydthetom*lm+dxdthetom*ym)
dnda1m = dxdthetom*ym - dyda1m*lm;
dndb1m = 0;                               %-dxdb1m*ym-dydb1m*lm
dndthetot = dxdb1m*ym + 2*dydb1m*lm;
%
% return to TLTRHOVR.M
```

185

```
% TLTRCRUS.M
% Follows  CTLTRGRP.M
% Computes the stability derivatives in cruise flight.
% calls the following subroutines
%
%
% APTRIM
% DCTPLOTS or DCTMATS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% STABILITY CALCULATION
%
aptrim
ctltrgrp
%
% computation of A,B,C,D matrices
%
g = 32.17;
m = GW/g;
%   CHECK THE WING CALCULATIONS
%
Amat=zeros(9);
Amat(1,1) = (dxdxdoth+dxdxdotv+dxdxdotf+dxdxdotw+dxdxdotm)/m;
Amat(1,2) = (dxdzdoth+dxdzdotf+dxdzdotw)/m;
Amat(1,3) = -wo;
Amat(1,4) = -g*cos(tho);
Amat(1,5) = (dxdydotv)/m;
Amat(1,8) = (dxdrw)/m+vo;              % recheck this one
%
Amat(2,1) = (dzdxdoth+dzdxdotf+dzdxdotw)/m;
Amat(2,2) = (dzdzdoth+dzdzdotf+dzdzdotw)/m;
Amat(2,3) = (dzdqh+dzdqw)/m+uo;
Amat(2,4) = -g*cos(pho)*sin(tho);
Amat(2,5) = (dzdydotw)/m;
Amat(2,6) = (dzdpw)/m-vo;
Amat(2,7) = -g*sin(pho)*cos(tho);
Amat(2,8) = (dzdrw)/m;                 % recheck this one
%
Amat(3,1) = (dmdxdoth+dmdxdotf+dmdxdotw+dmdxdotm)/Iyy;
Amat(3,2) = (dmdzdoth+dmdzdotf+dmdzdotw)/Iyy;
Amat(3,3) = (dmdqh+dmdqw)/Iyy;
Amat(3,8) = 0;                              %   dmdrf/Iyy;
%
Amat(4,3) = cos(pho);
Amat(4,8) = -sin(pho);
%
Amat(5,1) = (dydxdotv)/m;
Amat(5,4) = -g*sin(pho)*sin(tho);
Amat(5,5) = (dydydotv+dydydotf)/m;
Amat(5,6) = dydpv/m + wo;
Amat(5,7) = g*cos(pho)*cos(tho);
Amat(5,8) = (dydrv)/m - uo;
%
Amat(6,1) = (Izz*(drdxdotv)+Ixz*(dndxdotv))/Ic;
Amat(6,3) = 0;                                 % recheck this one
Amat(6,5) = (Izz*(drdydotv+drdydotf) + Ixz*(dndydotv+dndydotf))/Ic;
Amat(6,6) = (Izz*(drdpv+drdpw) + Ixz*(dndpv+dndpw))/Ic;
Amat(6,8) = (Izz*(drdrv+drdrw) + Ixz*(dndrw+dndrv))/Ic;
%
Amat(7,3) = sin(pho)*tan(tho);
```

186

```
Amat(7,6) = 1;
Amat(7,8) = cos(pho)*tan(tho);
%
Amat(8,1) = (Ixz*(drdxdotv) + Ixx*(dndxdotv))/Ic;
Amat(8,3) = 0;                              %  recheck
Amat(8,5) = (Ixz*(drdydotv+drdydotf) + Ixx*(dndydotv+dndydotf))/Ic;
Amat(8,6) = (Ixz*(drdpv+drdpw)+Ixx*(dndpw+dndpv))/Ic;
Amat(8,8) =(Ixz*(drdrv+drdrw)+Ixx*(dndrw+dndrv))/Ic;
Amat(9,8) = 1;
%                     longitudinal plant augmented is  X=[u w q theta]'
Flonaug=Amat(1:4,1:4);
Plonaug=poly(Flonaug);
Rlonaug=roots(Plonaug);
%                     lateral plant augmented is  X=[v p phi r psi]'
Flataug=Amat(5:9,5:9);
Plataug=poly(Flataug);
Rlataug=roots(Plataug);
%                     coupled plant
Pcoup=poly(Amat);
Rcoup=roots(Pcoup);
%
Bmat=zeros(9,4);
Bmat(1,2) = dxdthetom*dthetomddelc/m;
Bmat(1,3) = dxddela*ddeladlat/m;
Bmat(1,4) = dxddelr*ddelrddelp/m;
%
Bmat(2,1) = dzddele*ddeledlong/m;
Bmat(2,2) = dzdthetom*dthetomddelc/m;
%
Bmat(3,1) = dmddele*ddeledlong/Iyy;
Bmat(3,2) = dmdthetom*dthetomddelc/Iyy;
Bmat(3,3) = dmddela*ddeladlat/Iyy;
%
%
Bmat(5,1) = dyddele*ddeledlong/m;
Bmat(5,4) = dyddelr*ddelrddelp/m;
%
Bmat(6,3) = (Izz*drddela*ddeladlat + Ixz*dnddela*ddeladlat)/Ic;
Bmat(6,4) = (Izz*drddelr*ddelrddelp + Ixz*dnddelr*ddelrddelp)/Ic;
%
Bmat(8,3) = (Ixz*drddela*ddeladlat + Ixx*dnddela*ddeladlat)/Ic;
Bmat(8,4) = (Ixz*drddelr*ddelrddelp + Ixx*dnddelr*ddelrddelp)/Ic;
%
Glonaug = Bmat(1:4,1:4);
Glataug = Bmat(5:9,1:4);
% coupled input matrix
%
%xcouple=12/lockno*e/R/(1+e/3/R);
% designed damping
desdmdq=dmdqh+dmdqw;
desdrdp=drdpw+drdpv;
desdndr=dndrv+dndrw;
% cooper harper pilot rating
prpitch=desdmdq/Iyy;
prroll=desdrdp/Ixx;
pryaw=desdndr/Izz;
% control power
cppitch = Bmat(3,1)*Iyy;
cproll = Bmat(6,3)*Ixx;
cpyaw = Bmat(8,4)*Izz;
```

187

```
cpipitch = Bmat(3,1);
cpiroll = Bmat(6,3);
cpiyaw = Bmat(8,4);
%
%theta0=theta7;
```

```
% TLTRHOVR.M
% CALLED BY STAB.M
% Computes the stability derivatives at a hover for a tilt rotor.
% calls the FOLLOWING subroutines to compute stability derivatives
%
% HTLTRGRP.M
% TILTTRIM.M
%
% computation of stability derivatives
% the only derivatives important at hover are main rotors
format compact
% evaluate dctsigdtheto dcqsigdtheto dctsigdthetot and dcqsigdthetot
%
hvrtrm15
htltrgrp
Ic = Ixx*Izz-Ixz^2;
%
% computation of A & B matrices
%
Amat=zeros(9);
Amat(1,1) = (dxdxdotm)/m;
Amat(1,2) = (dxdzdotm)/m;
Amat(1,3) = (dxdqm)/m-wo;
Amat(1,4) = -g*cos(tho);
Amat(1,5) = (dxdydotm)/m;
Amat(1,6) = (dxdpm)/m;
Amat(1,8) = (dxdrm)/m+vo;
%
Amat(2,1) = (dzdxdotm)/m;
Amat(2,2) = (dzdzdotm)/m;
Amat(2,3) = (dzdqm)/m+uo;
Amat(2,4) = -g*cos(pho)*sin(tho);
Amat(2,5) = (dzdydotm)/m;
Amat(2,6) = (dzdpm)/m-vo;
Amat(2,7) = -g*sin(pho)*cos(tho);
Amat(2,8) = (dzdrm)/m;
%
Amat(3,1) = (dmdxdotm)/Iyy;
Amat(3,2) = (dmdzdotm)/Iyy;
Amat(3,3) = (dmdqm)/Iyy;
Amat(3,5) = (dmdydotm)/Iyy;
Amat(3,6) = (dmdpm)/Iyy;
Amat(3,8) = (dmdrm)/Iyy;
%
Amat(4,3) = cos(pho);
Amat(4,8) = -sin(pho);
%
Amat(5,1) = (dydxdotm)/m;
Amat(5,2) = (dydzdotm)/m;
Amat(5,3) = (dydqm)/m;
Amat(5,4) = -g*sin(pho)*sin(tho);
Amat(5,5) = (dydydotm)/m;
Amat(5,6) = (dydpm)/m+wo;
Amat(5,7) = g*cos(pho)*cos(tho);
Amat(5,8) = (dydrm)/m-uo;
%
Amat(6,1) = (Izz*(drdxdotm)+Ixz*(dndxdotm))/Ic;
Amat(6,2) = (Izz*(drdzdotm)+Ixz*(dndzdotm))/Ic;
Amat(6,3) = (Izz*(drdqm)+Ixz*(dndqm))/Ic;
Amat(6,5) = (Izz*(drdydotm)+Ixz*(dndydotm))/Ic;
```

189

```
Amat(6,6) = (Izz*(drdpm)+Ixz*(dndpm))/Ic;
Amat(6,8) = (Izz*(drdrm)+Ixz*(dndrm))/Ic;
%
Amat(7,3) = sin(pho)*tan(tho);
Amat(7,6) = 1;
Amat(7,8) = cos(pho)*tan(tho);
%
Amat(8,1) = (Ixz*(drdxdotm)+Ixx*(dndxdotm))/Ic;
Amat(8,2) = (Ixz*(drdzdotm)+Ixx*(dndzdotm))/Ic;
Amat(8,3) = (Ixz*(drdqm)+Ixx*(dndqm))/Ic;
Amat(8,5) = (Ixz*(drdydotm)+Ixx*(dndydotm))/Ic;
Amat(8,6) = (Ixz*(drdpm)+Ixx*(dndpm))/Ic;
Amat(8,8) = (Ixz*(drdrm)+Ixx*(dndrm))/Ic;
%
Amat(9,8) = 1;
% longitudinal plant augmented X=[u w q theta]
Flonaug = Amat(1:4,1:4);
Plonaug=poly(Flonaug);
Rlonaug=roots(Plonaug);
%
% Lateral plant augmented with X=[v p phi r psi]
Flataug = Amat(5:9,5:9);
Plataug=poly(Flataug);
Rlataug=roots(Plataug);
%                              coupled plant
Pcoup=poly(Amat);
Rcoup=roots(Pcoup);
%
Bmat = zeros(9,4);
Bmat(1,1) = (dxdb1m*db1mddele)/m;
Bmat(1,2) = dxdthetom*dthetomddelc/m;
Bmat(1,3) = 0;                              %dxda1m*da1mddela/m;
Bmat(1,4) = 0;                     %(dxdb1m*(db1mddele+dthetotddelp))/m;
%
Bmat(2,1) = dzdb1m*db1mddele/m;
Bmat(2,2) = dzdthetom*dthetomddelc/m;
Bmat(2,3) = 0;                              %dzda1m*da1mddela/m;
Bmat(2,4) = 0;                     %dzdb1m*(db1mddele+dthetotddelp)/m;
%
Bmat(3,1) = dmdb1m*db1mddele/Iyy;
Bmat(3,2) = dmdthetom*dthetomddelc/Iyy;
Bmat(3,3) = 0;                     %dmda1m*da1mddela/Iyy;
Bmat(3,4) = 0;                     %dmdb1m*(db1mddele+dthetotddelp)/Iyy;
%
%
Bmat(5,1) = 0;                     %dydb1m*(db1mddele-dthetotddelp)/m;
Bmat(5,2) = 0;
%dydthetom*(dthetomddelc+dthetomddela)/m;
Bmat(5,3) = dyda1m*da1mddela/m;
Bmat(5,4) = dydthetot*dthetotddelp/m;
%
Bmat(6,1) = 0;
%(Izz*drdb1m*(db1mddele-dthetotddelp)+Ixz*dndb1m*(db1mddele-dthetotddelp))/Ic;
Bmat(6,2) = 0;
%(Izz*drdthetom*(dthetomddelc+dthetomddela)+Ixz*dndthetom*(dthetomddelc+dthetom
ddela))/Ic;
Bmat(6,3) = (Izz*drda1m*da1mddela+Ixz*dnda1m*da1mddela)/Ic;
Bmat(6,4) = (Izz*drdthetot*dthetotddelp+Ixz*dndthetot*dthetotddelp)/Ic;
%
%
```

190

```
Bmat(8,1) = 0;
%(Ixz*drdb1m*(db1mddele-dthetotddelp)+Ixx*dndb1m*(db1mddele-dthetotddelp))/Ic;
Bmat(8,2) = 0;
%(Ixz*drdthetom*(dthetomddelc+dthetomddela)+Ixx*dndthetom*(dthetomddelc+dthetom
ddela))/Ic;
Bmat(8,3) = (Ixz*drda1m*da1mddela+Ixx*dnda1m*da1mddela)/Ic;
Bmat(8,4) = (Ixz*drdthetot*dthetotddelp+Ixx*dndthetot*dthetotddelp)/Ic;
%
Glonaug = Bmat(1:4,1:4);
%
Glataug = Bmat(5:9,1:4);
% coupled input matrix
% cross coupling
xcouple=12/lockno*e/R/(1+e/3/R);
% designed damping
desdmdq=dmdqm;
desdrdp = drdpm;                           %+drdpt+drdpn
desdndr = dndrm;                           %+dndrt+dndrn
% now cooper harper pilot rating
prpitch=desdmdq/Iyy;
prroll=(drdpm)/Ixx;                        %+drdpt+drdpn
pryaw=desdndr/Izz;
% control power
cppitch=Bmat(3,1)*Iyy;
cproll=Bmat(6,3)*Ixx;
cpyaw=Bmat(8,4)*Izz;
cpipitch=Bmat(3,1);
cpiroll=Bmat(6,3);
cpiyaw=Bmat(8,4);
%
%thetao=theta7;
```

191

```
% TMRESPC.M
% open loop time response for longitudinal and lateral plants %
disp('While viewing a plot, press any key to go to the next plot')
disp(' ')
disp('Do you want to see longitudinal or lateral/directional plots?')
disp(' ')
disp('1.  Longitudinal (eight plots total).')
disp('2.  Lateral Directional (ten plots total).')
disp(' ')
pview=input('Enter a number : ');
clc
T = linspace(0,10,100);
Du=[0 0 0 0];
if pview==1
  U = zeros(length(T),4);
  C = [0 0 0 0];
  disp('longitudinal cylic 0.5 sec (+1") pulse')
% command time response to dele pulse
  U(:,1) = stepfun(T',.5) - stepfun(T',1);
  [Y,X] = lsim(Flonaug,Glonaug,C,Du,U,T);
  plot(T,X(:,1),'w')
  title('U Response to Longitudinal Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Forward Velocity, U (ft/sec)')
  pause
  %!del tre2uc.*
  print -dmeta tre2uc
% command time response of e to w
  plot(T,X(:,2),'w')
  title('W Response to Longitudinal Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Vertical Velocity, W (ft/sec)')
  pause
  %!del tre2wc.*
  print -dmeta tre2wc
% command time response of e to q
  plot(T,X(:,3),'w')
  title('Pitch Rate Response to Longitudinal Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Pitch Rate, q (rad/sec)')
  pause
  %!del tre2qc.*
  print -dmeta tre2qc
% command time response of e to theta
  plot(T,X(:,4),'w')
  title('Pitch Angle Response to Longitudinal Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Pitch Angle, theta (rad)')
  pause
  %!del tre2thec.*
  print -dmeta tre2thec
%
% now collective
  disp('collective step')
% command time response to delc step
  U = zeros(length(T),4);
  U(:,2) = stepfun(T',.5);
  [Y,X] = lsim(Flonaug,Glonaug,C,Du,U,T);
  plot(T,X(:,1),'w')
  title('U Response to Collective step , Cruise')
```

192

```
    xlabel('Time, seconds')
    ylabel('Forward Velocity, U (ft/sec)')
    pause
    %!del trc2uc.*
    print -dmeta trc2uc
% command time response of c to w
    plot(T,X(:,2),'w')
    title('W Response to Collective step , Cruise')
    xlabel('Time, seconds')
    ylabel('Vertical Velocity, W (ft/sec)')
    pause
    %!del trc2wc.*
    print -dmeta trc2wc
% command time response of e to q
    plot(T,X(:,3),'w')
    title('Pitch Rate Response to Collective step, Cruise')
    xlabel('Time, seconds')
    ylabel('Pitch Rate, q (rad/sec)')
    pause
    %!del trc2qc.*
    print -dmeta trc2qc
% command time response of e to theta
    plot(T,X(:,4),'w')
    title('Pitch Angle Response to Collective step, Cruise')
    xlabel('Time, seconds')
    ylabel('Pitch Angle, theta (rad)')
    pause
    %!del trc2thec.*
    print -dmeta trc2thec
    clc
    disp(' ')
    disp('Plots are saved under the following filenames:')
    disp(' ')
    disp('Longitudinal Cyclic')
    disp('Longitudinal Cyclic to U, Cruise - tre2uc.wmf')
    disp('Longitudinal Cyclic to Theta, Cruise - tre2thec.wmf')
    disp('Longitudinal Cyclic to Pitch Rate, Cruise - tre2qc.wmf')
    disp('Longitudinal Cyclic to W, Cruise - tre2wc.wmf')
    disp('')
    disp('Collective')
    disp('Collective to U, Cruise - trc2uc.wmf')
    disp('Collective to Pitch, Cruise - trc2thec.wmf')
    disp('Collective to Pitch Rate, Cruise - trc2qc.wmf')
    disp('Collective to W, Cruise - trc2wc.wmf')
    disp(' ')
    disp('press any key to continue ...')
    pause
%
% now for lateral directional plant
elseif pview==2
%
% now for lateral directional plant
    C = [0 0 0 0 0];
    U = zeros(length(T),4);
% lateral cyclic pulse
    disp('lateral cylic 0.5 sec (+1") pulse')
% command time response to dela pulse
    U(:,3) = stepfun(T',.5) - stepfun(T',1);
    [Y,X] = lsim(Flataug,Glataug,C,Du,U,T);
% command time response of dela to V
```

193

```
  plot(T,X(:,1),'w')
  title('V Response to Lateral Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Sideward Velocity, V (ft/sec)')
  pause
  %!del tra2vc.*
  print -dmeta tra2vc
% command time response of dela to p
  plot(T,X(:,2),'w')
  title('Roll Rate Response to Lateral Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Roll Rate, p (rad/sec)')
  pause
  %!del tra2pc.*
  print -dmeta tra2pc
% command time response of dela to phi
  plot(T,X(:,3),'w')
  title('Roll Angle Response to Lateral Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Roll Angle, phi (ft/sec)')
  pause
  %!del tra2phic.*
  print -dmeta tra2phic
% command time response of dela to r
  plot(T,X(:,4),'w')
  title('Yaw Rate Response to Lateral Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Yaw Rate, r (rad/sec)')
  pause
  %!del tra2rc.*
  print -dmeta tra2rc
% command time response of dela to Psi
  plot(T,X(:,5),'w')
  title('Yaw Angle Response to Lateral Cyclic Pulse , Cruise')
  xlabel('Time, seconds')
  ylabel('Yaw Angle, Psi (rad/sec)')
  pause
  %!del tra2yc.*
  print -dmeta tra2yc
%
% pedals
  U = zeros(length(T),4);
% directional pedals doublet
  disp('directional pedals doublet 1.5 sec (+/- 1") pulse')
% command time response to delp doublet
  U(:,4) = stepfun(T',.5) - 2*stepfun(T',1) + stepfun(T',1.5);
  [Y,X] = lsim(Flataug,Glataug,C,Du,U,T);
% command time response of delp to V
  plot(T,X(:,1),'w')
  title('V Response to Directional Pedal Doublet , Cruise')
  xlabel('Time, seconds')
  ylabel('Sideward Velocity, V (ft/sec)')
  pause
  %!del tra2vc.*
  print -dmeta trp2vc
% command time response of delp to p
  plot(T,X(:,2),'w')
  title('Roll Rate Response to Directional Pedal Doublet , Cruise')
  xlabel('Time, seconds')
  ylabel('Roll Rate, p (rad/sec)')
```

194

```
    pause
    %!del trp2pc.*
    print -dmeta trp2pc
% command time response of delp to phi
    plot(T,X(:,3),'w')
    title('Roll Angle Response to Directional Pedal Doublet , Cruise')
    xlabel('Time, seconds')
    ylabel('Roll Angle, phi (ft/sec)')
    pause
    %!del trp2phic.*
    print -dmeta trp2phic
% command time response of delp to r
    plot(T,X(:,4),'w')
    title('Yaw Rate Response to Directional Pedal Doublet , Cruise')
    xlabel('Time, seconds')
    ylabel('Yaw Rate, r (rad/sec)')
    pause
    %!del trp2rc.*
    print -dmeta trp2rc
% command time response of delp to Psi
    plot(T,X(:,5),'w')
    title('Yaw Angle Response to Directional Pedal Doublet , Cruise')
    xlabel('Time, seconds')
    ylabel('Yaw Angle, Psi (rad/sec)')
    pause
    %!del trp2yc.*
    print -dmeta trp2yc
    clc
    disp(' ')
    disp('Plots are saved under the following filenames:')
    disp(' ')
    disp('Lateral cyclic')
    disp('Lateral Cyclic to Bank, Cruise - tra2phic.wmf')
    disp('Lateral Cyclic to Sideslip (v), Cruise - tra2vc.wmf')
    disp('Lateral Cyclic to Roll Rate, Cruise - tra2pc.wmf')
    disp('Lateral Cyclic to Yaw Rate, Cruise - tra2rc.wmf')
    disp('Lateral Cyclic to Yaw, Cruise - tra2yc.wmf')
    disp('Pedals')
    disp('Pedals to Bank, Cruise - trp2phic.wmf')
    disp('Pedals to Sideslip (v), Cruise - trp2vc.wmf')
    disp('Pedals to Roll Rate, Cruise - trp2pc.wmf')
    disp('Pedals to Yaw Rate, Cruise - trp2rc.wmf')
    disp('Pedals to Yaw, Cruise - trp2yc.wmf')
    disp(' ')
    disp('press any key to continue ...')
    pause
%
end
```

```
% TMRESPH.M
% open loop time response for longitudinal and lateral plants %
disp('While viewing a plot, press any key to go to the next plot')
disp(' ')
disp('Do you want to see longitudinal or lateral/directional plots?')
disp(' ')
disp('1.   Longitudinal (eight plots total).')
disp('2.   Lateral Directional (ten plots total).')
disp(' ')
pview=input('Enter a number : ');
clc
T = linspace(0,10,100);
Du=[0 0 0 0];
if pview==1
  U = zeros(length(T),4);
  C = [0 0 0 0];
  disp('longitudinal cylic 0.5 sec (+1") pulse')
% command time response to dele pulse
  U(:,1) = stepfun(T',.5) - stepfun(T',1);
  [Y,X] = lsim(Flonaug,Glonaug,C,Du,U,T);
  plot(T,X(:,1),'w')
  title('U Response to Longitudinal Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Forward Velocity, U (ft/sec)')
  pause
  %!del tre2uh.*
  print -dmeta tre2uh
% command time response of e to w
  plot(T,X(:,2),'w')
  title('W Response to Longitudinal Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Vertical Velocity, W (ft/sec)')
  pause
  %!del tre2wh.*
  print -dmeta tre2wh
% command time response of e to q
  plot(T,X(:,3),'w')
  title('Pitch Rate Response to Longitudinal Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Pitch Rate, q (rad/sec)')
  pause
  %!del tre2qh.*
  print -dmeta tre2qh
% command time response of e to theta
  plot(T,X(:,4),'w')
  title('Pitch Angle Response to Longitudinal Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Pitch Angle, theta (rad)')
  pause
  %!del tre2theh.*
  print -dmeta tre2theh
%
% now collective
  disp('collective step')
% command time response to delc step
  U = zeros(length(T),4);
  U(:,2) = stepfun(T',.5);
  [Y,X] = lsim(Flonaug,Glonaug,C,Du,U,T);
  plot(T,X(:,1),'w')
  title('U Response to Collective step , Hover')
```

196

```matlab
  xlabel('Time, seconds')
  ylabel('Forward Velocity, U (ft/sec)')
  pause
  %!del trc2uh.*
  print -dmeta trc2uh
% command time response of c to w
  plot(T,X(:,2),'w')
  title('W Response to Collective step , Hover')
  xlabel('Time, seconds')
  ylabel('Vertical Velocity, W (ft/sec)')
  pause
  %!del trc2wh.*
  print -dmeta trc2wh
% command time response of e to q
  plot(T,X(:,3),'w')
  title('Pitch Rate Response to Collective step, Hover')
  xlabel('Time, seconds')
  ylabel('Pitch Rate, q (rad/sec)')
  pause
  %!del trc2qh.*
  print -dmeta trc2qh
% command time response of e to theta
  plot(T,X(:,4),'w')
  title('Pitch Angle Response to Collective step, Hover')
  xlabel('Time, seconds')
  ylabel('Pitch Angle, theta (rad)')
  pause
  %!del trc2theh.*
  print -dmeta trc2theh
  clc
  disp(' ')
  disp('Plots are saved under the following filenames:')
  disp(' ')
  disp('Longitudinal Cyclic')
  disp('Longitudinal Cyclic to U, Hover - tre2uh.wmf')
  disp('Longitudinal Cyclic to Theta, Hover - tre2theh.wmf')
  disp('Longitudinal Cyclic to Pitch Rate, Hover - tre2qh.wmf')
  disp('Longitudinal Cyclic to W, Hover - tre2wh.wmf')
  disp('')
  disp('Collective')
  disp('Collective to U, Hover - trc2uh.wmf')
  disp('Collective to Pitch, Hover - trc2theh.wmf')
  disp('Collective to Pitch Rate, Hover - trc2qh.wmf')
  disp('Collective to W, Hover - trc2wh.wmf')
  disp(' ')
  disp('press any key to continue ...')
  pause
%
% now for lateral directional plant
elseif pview==2
%
% now for lateral directional plant
  C = [0 0 0 0 0];
  U = zeros(length(T),4);
% lateral cyclic pulse
  disp('lateral cylic 0.5 sec (+1") pulse')
% command time response to dela pulse
  U(:,3) = stepfun(T',.5) - stepfun(T',1);
  [Y,X] = lsim(Flataug,Glataug,C,Du,U,T);
% command time response of dela to V
```

```
  plot(T,X(:,1),'w')
  title('V Response to Lateral Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Sideward Velocity, V (ft/sec)')
  pause
  %!del tra2vh.*
  print -dmeta tra2vh
% command time response of dela to p
  plot(T,X(:,2),'w')
  title('Roll Rate Response to Lateral Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Roll Rate, p (rad/sec)')
  pause
  %!del tra2ph.*
  print -dmeta tra2ph
% command time response of dela to phi
  plot(T,X(:,3),'w')
  title('Roll Angle Response to Lateral Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Roll Angle, phi (ft/sec)')
  pause
  %!del tra2phih.*
  print -dmeta tra2phih
% command time response of dela to r
  plot(T,X(:,4),'w')
  title('Yaw Rate Response to Lateral Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Yaw Rate, r (rad/sec)')
  pause
  %!del tra2rh.*
  print -dmeta tra2rh
% command time response of dela to Psi
  plot(T,X(:,5),'w')
  title('Yaw Angle Response to Lateral Cyclic Pulse , Hover')
  xlabel('Time, seconds')
  ylabel('Yaw Angle, Psi (rad/sec)')
  pause
  %!del tra2psih.*
  print -dmeta tra2psih
%
% pedals
  U = zeros(length(T),4);
% directional pedals doublet
  disp('directional pedals doublet 1.5 sec (+/- 1") pulse')
% command time response to delp doublet
  U(:,4) = stepfun(T',.5) - 2*stepfun(T',1) + stepfun(T',1.5);
  [Y,X] = lsim(Flataug,Glataug,C,Du,U,T);
% command time response of delp to V
  plot(T,X(:,1),'w')
  title('V Response to Directional Pedal Doublet , Hover')
  xlabel('Time, seconds')
  ylabel('Sideward Velocity, V (ft/sec)')
  pause
  %!del trp2vh.*
  print -dmeta trp2vh
% command time response of delp to p
  plot(T,X(:,2),'w')
  title('Roll Rate Response to Directional Pedal Doublet , Hover')
  xlabel('Time, seconds')
  ylabel('Roll Rate, p (rad/sec)')
```

198

```
   pause
   %!del trp2ph.*
   print -dmeta trp2ph
% command time response of delp to phi
   plot(T,X(:,3),'w')
   title('Roll Angle Response to Directional Pedal Doublet , Hover')
   xlabel('Time, seconds')
   ylabel('Roll Angle, phi (ft/sec)')
   pause
   %!del trp2phih.*
   print -dmeta trp2phih
% command time response of delp to r
   plot(T,X(:,4),'w')
   title('Yaw Rate Response to Directional Pedal Doublet , Hover')
   xlabel('Time, seconds')
   ylabel('Yaw Rate, r (rad/sec)')
   pause
   %!del trp2rh.*
   print -dmeta trp2rh
% command time response of delp to Psi
   plot(T,X(:,5),'w')
   title('Yaw Angle Response to Directional Pedal Doublet , Hover')
   xlabel('Time, seconds')
   ylabel('Yaw Angle, Psi (rad/sec)')
   pause
   %!del trp2psih.*
   print -dmeta trp2psih
   clc
   disp(' ')
   disp('Plots are saved under the following filenames:')
   disp(' ')
   disp('Lateral cyclic')
   disp('Lateral Cyclic to Bank, Hover - tra2phih.wmf')
   disp('Lateral Cyclic to Sideslip (v), Hover - tra2vh.wmf')
   disp('Lateral Cyclic to Roll Rate, Hover - tra2ph.wmf')
   disp('Lateral Cyclic to Yaw Rate, Hover - tra2rh.wmf')
   disp('Lateral Cyclic to Yaw, Hover - tra2psih.wmf')
   disp('Pedals')
   disp('Pedals to Bank, Hover - trp2phih.wmf')
   disp('Pedals to Sideslip (v), Hover - trp2vh.wmf')
   disp('Pedals to Roll Rate, Hover - trp2ph.wmf')
   disp('Pedals to Yaw Rate, Hover - trp2rh.wmf')
   disp('Pedals to Yaw, Hover - trp2psih.wmf')
   disp(' ')
   disp('press any key to continue ...')
   pause
%
end
```

LIST OF REFERENCES

1. Etkin, Bernard, *Dynamics of Flight --- Stability and Control*, 2nd Ed., John Wiley and Sons, NY, 1982.

2. Ferguson, Samuel W., "A Mathmatical Model for Real Time Flight Simulation of a Generic Tilt-Rotor Aircraft", NASA Contractor Report CR-166536 (Technical Report No. 1195-2), Systems Technology, Inc., Mountain View, CA , 1988.

3. Ferguson, Samuel W., "Development and Validation of a Simulation for a Generic Tilt-Rotor Aircraft", NASA Contractor Report CR-166537 (Technical Report No. 1195-1), Systems Technology, Inc., Mountain View, CA , 1989.

4. Ferguson, Samuel W., GTRS model simulation results from a dedicated run , EMA, Mansfield, TX, 1995.

5. Nicolai, Leland M, *Fundamentals of Aircraft Design*, METS,Inc, San Jose, 1984.

6. Prouty, Raymond. W., *Helicopter Performance, Stabilty and Control*, R.E. Krieger Publ. Co., Malabar, FL, 1990.

7. Wirth, Walter , "Linear Modeling of Rotorcraft for Stability Analysis and Preliminary Design", Master's Thesis, Naval Postgraduate School, Monterey, CA., 1993.

8. Talbot, Peter D., Bruce E. Tinling, William A. Decker, and Robert N. Chen, "A Mathematical Model of a Single Main Rotor Helicopter for Piloted Simulation", NASA Technical Memornadum 84281, Ames Research Center, Moffet Field, CA, 1982.

9. Eccles, David M., "A Validation of the Joint Army/Navy Rotorcraft Analysis and Design Software by Comparison with H-34 and UH-60A Flight Test", Master's Thesis, Naval Postgraduate School, Monterey, CA., 1993.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Road
   Quantico, Virginia 22134-5027

4. Director, Marine Corps Research Center 2
   MCCDC, Code: C40RC
   2040 Broadway Street
   Quantico, Virginia 22134-5107

5. Director, Studies and Analysis Division 1
   MCCDC, Code C45
   3300 Russell Road
   Quantico, Virginia 22134-5130

6. Chairman, Code AA 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5106

7. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943-5121

8. Professor E. Roberts Wood, Code AA/Wd 4
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5106